PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-359616

(43)Date of publication of application : 13.12.2002

---

(51)Int.Cl. H04L 9/08
G09C 5/00
H04L 9/00
H04L 9/32

---

(21)Application number : 2002-028915 (71)Applicant : SONY CORP

(22)Date of filing : 06.02.2002 (72)Inventor : TANAKA KOICHI
KAWAKAMI TATSU
KURODA HISASUKE
ISHIGURO RYUJI

---

(30)Priority
Priority number : 2001033114
2001094803
Priority date : 09.02.2001
29.03.2001
Priority country : JP
JP

---

(54) INFORMATION PROCESSOR AND METHOD, LICENSE SERVER, AND PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To freely distribute contents and allow only authorized users to utilize the contents.
SOLUTION: A client receives an encrypted content from a content server. The header of the content includes license-identifying information for identifying a license required in utilization of the content. The client requests a license server to transmit a license identified by the license-identifying information. When receiving the request for a license, the license server carries out a charging process before transmitting the license to the client. The client can decode and play back the content on the condition of possessing the license.

LEGAL STATUS [Date of request for examination] 20.01.2005
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

CLAIMS

[Claim(s)]
[Claim 1] In the information processor which permits utilization of contents a condition [holding the license] The license specific information for specifying said license which carries out utilization authorization of the contents concerned, A contents storage means to memorize said contents including the enciphered contents data and key information required in order to decode contents data, A license storage means to memorize the license containing the contents specific information for specifying said contents by which utilization authorization is carried out, A judgment means to judge whether the license which can carry out utilization authorization of said contents is memorized by said license storage means, The information processor characterized by having a decode means to decode the contents data of said contents a condition [it having been judged that the license was memorized by said judgment means].
[Claim 2] It is the information processor according to claim 1 carry out that the license which said information processor was equipped with a transmitting means to transmit the license demand which contains the license identification information for identifying a license in a license server further, and a receiving means to receive the license transmitted by the license server, and was received by said receiving means is memorized by said license storage means as the description.
[Claim 3] Said contents data are an information processor according to claim 1 characterized by having further a playback means to reproduce the contents data which are data which combined text data, image data, voice data, a video data, or them, and were decoded by said decode means.
[Claim 4] It is the information processor according to claim 1 which said key

information is equipped with a device node key storage means to by_which said information processor memorizes a device node key further, including EKB (Enabling Key Block), and is characterized by for said decode means to decode said enciphered contents data using the root key by which decode processing might be carried out in said EKB (Enabling Key Block) using said device node key memorized by said device node key storage means.

[Claim 5] Said key information contains the contents key further enciphered by said root key of EKB (Enabling Key Block). Said contents data are enciphered by said contents key. Said decode means said contents key decoded using the root key by which decode processing might be carried out in said EKB (Enabling Key Block) using said device node key memorized by said device node key storage means The information processor according to claim 4 characterized by using and decoding said enciphered contents data.

[Claim 6] Said license is an information processor according to claim 1 characterized by including the service-condition information which shows further the service condition of contents which becomes available according to the license concerned.

[Claim 7] Said license is an information processor according to claim 1 characterized by including further the electronic signature made with the private key of a license server.

[Claim 8] Said information processor is equipped with a terminal identification information storage means to memorize the terminal identification information which identifies an information processor further. Said license demand includes further said terminal identification information memorized by the terminal identification information storage means. Said license received by said receiving means includes said terminal identification information further. Said judgment means Said terminal identification information memorized by said terminal identification information included in said license and said terminal identification information storage means is compared. The information processor according to claim 2 characterized by restricting when both are in agreement, and judging that the license concerned is a license to which utilization of said contents is permissible.

[Claim 9] The license specific information for specifying said license which is the information processing approach of permitting utilization of contents a condition [holding the license], and carries out utilization authorization of the contents concerned, The enciphered contents data and key information required in order to decode contents data, The step which memorizes the license containing the contents specific information for specifying the step which memorizes ****** contents, and

said contents in which utilization authorization is carried out by the license concerned, The step which judges whether the license which can carry out utilization authorization of said contents is memorized by said license storage means, The information processing approach characterized by including the step which decodes the contents data of said contents a condition [it having been judged that the license was memorized by said judgment means].

[Claim 10] The license specific information for being the program which makes a computer perform processing to which utilization of contents is permitted a condition [holding the license], and specifying said license which carries out utilization authorization of the contents concerned, The enciphered contents data and key information required in order to decode contents data, The step which memorizes the license containing the contents specific information for specifying the step which memorizes ****** contents, and said contents in which utilization authorization is carried out by the license concerned, The step which judges whether the license which can carry out utilization authorization of said contents is memorized by said license storage means, The program which makes a computer perform the step which decodes the contents data of said contents a condition [it having been judged that the license was memorized by said judgment means].

[Claim 11] The program according to claim 10 characterized by enciphering said program or its part.

[Claim 12] In the license server which publishes the license to which utilization of contents is permitted The contents specific information for specifying said contents in which utilization authorization is carried out by the license concerned, A receiving means transmitted from a license storage means to memorize said license including the terminal identification information which identifies an information processor, and the information processor to receive the license demand containing the license identification information which identifies a license, An extract means to extract said license corresponding to said license identification information contained in said license demand from said license storage means, A processing means to add said terminal identification information to said license extracted by said extract means, A signature means to add electronic signature to the license to which terminal identification information was added by said processing means using the private key of a license server, The license server characterized by having a transmitting means to transmit the license signed by said signature means to the information processor which transmitted said license demand.

[Claim 13] The contents specific information for specifying said contents in which are

the information processing approaches of publishing the license to which utilization of contents is permitted, and utilization authorization is carried out by the license concerned, The step which memorizes said license including the terminal identification information which identifies an information processor, The step which receives the license demand containing the license identification information which was transmitted from the information processor, and which identifies a license, The step which extracts said license corresponding to said license identification information contained in said license demand from said license storage means, The step which adds said terminal identification information to said license extracted by said extract means, The step which adds electronic signature to the license to which terminal identification information was added by said processing means using the private key of a license server, The information processing approach characterized by including the step which transmits the license signed by said signature means to the information processor which transmitted said license demand.

[Claim 14] The contents specific information for being the program which makes a computer perform processing processing which publishes the license to which utilization of contents is permitted, and specifying said contents in which utilization authorization is carried out by the license concerned, The step which memorizes said license including the terminal identification information which identifies an information processor, The step which receives the license demand containing the license identification information which was transmitted from the information processor, and which identifies a license, The step which extracts said license corresponding to said license identification information contained in said license demand from said license storage means, The step which adds said terminal identification information to said license extracted by said extract means, The step which adds electronic signature to the license to which terminal identification information was added by said processing means using the private key of a license server, The program which makes a computer perform the step which transmits the license signed by said signature means to the information processor which transmitted said license demand.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] It is related with a program at the information processor and the approach the contents to which especially this invention is not licensed in an information processor and an approach, a license server, and a list from a copyright person about the program enabled it to prevent it being copied unjustly and used, a license server, and a list.

[0002]

[Description of the Prior Art] Recently, as a user provides other users with the music data which he holds through the Internet and offer is received for the music data which he does not hold from other users, the system which two or more users exchange music data for nothing, and suit is realized.

[0003] If the contents of one music and others exist, since other users become [no] possible [using it], many users stop purchasing contents and the copyright person about contents cannot sell the contents as a work at such a system theoretically, an opportunity to receive the loyalty about utilization of the work which can originally be received will be lost with a sale of a work.

[0004]

[Problem(s) to be Solved by the Invention] Then, it is requested socially that it should prevent being used unjustly, without barring the negotiation of contents.

[0005] This invention is made in view of such a situation, and it enables it to prevent certainly that contents are used unjustly.

[0006]

[Means for Solving the Problem] The license specific information for specifying a license required in order that the information processor of this invention may carry out utilization authorization of the contents, A contents storage means to memorize

contents including the enciphered contents data and key information required in order to decode contents data, A license storage means to memorize the license containing the contents specific information for specifying the contents by which utilization authorization is carried out, A judgment means to judge whether the license which can carry out utilization authorization of the contents is memorized by the license storage means, It is characterized by having a decode means to decode the contents data of contents a condition [it having been judged that the license was memorized by the judgment means].

[0007] An information processor has a transmitting means to transmit the license demand which contains the license identification information for identifying a license in a license server further, and a receiving means to receive the license transmitted by the license server, and the license received by the receiving means can be memorized by the license storage means.

[0008] Contents data are data which combined text data, image data, voice data, a video data, or them, and can be further equipped with a playback means to reproduce the contents data decoded by the decode means.

[0009] Key information is equipped with a device node key storage means by which an information processor memorizes a device node key further, including EKB (Enabling Key Block), and a decode means can decode the contents data enciphered using the root key by which decode processing might be carried out in EKB (EnablingKey Block) using the device node key memorized by the device node key storage means.

[0010] Including the contents key as which key information was further enciphered by the root key of EKB (Enabling Key Block), contents data are enciphered by the contents key and a decode means can decode the contents data enciphered using the contents key decoded using the root key by which decode processing might be carried out in EKB (Enabling Key Block) using the device node key memorized by the device node key storage means.

[0011] A license can include the service-condition information which shows further the service condition of contents which becomes available according to the license.

[0012] A license can include further the electronic signature made with the private key of a license server.

[0013] An information processor is equipped with a terminal identification information storage means to memorize the terminal identification information which identifies an information processor further. The license with which the license demand was received by the receiving means including the terminal identification information further memorized by the terminal identification information storage means includes

terminal identification information further. A judgment means The terminal identification information memorized by the terminal identification information included in a license and the terminal identification information storage means is compared, when both are in agreement, it restricts, and it can judge that the license is a license to which utilization of contents is permissible.

[0014] License specific information for the information processing approach of this invention to specify the license which carries out utilization authorization of the contents, The enciphered contents data and key information required in order to decode contents data, The step which memorizes the step which memorizes ****** contents, and the license containing the contents specific information for specifying the contents by which utilization authorization is carried out, The step which judges whether the license which can carry out utilization authorization of the contents is memorized by the license storage means, It is characterized by including the step which decodes the contents data of contents a condition [it having been judged that the license was memorized by the judgment means].

[0015] License specific information for the program of this invention to specify the license which carries out utilization authorization of the contents, The enciphered contents data and key information required in order to decode contents data, The step which memorizes the step which memorizes ****** contents, and the license containing the contents specific information for specifying the contents by which utilization authorization is carried out, The step which judges whether the license which can carry out utilization authorization of the contents is memorized by the license storage means, A computer is made to perform the step which decodes the contents data of contents a condition [it having been judged that the license was memorized by the judgment means].

[0016] A program or its part can be enciphered.

[0017] Contents specific information for the license server of this invention to specify the contents permitted, A license storage means to memorize the license including the terminal identification information which identifies an information processor, A receiving means to receive the license demand containing the license identification information which was transmitted from the information processor and which identifies a license, An extract means to extract the license corresponding to the license identification information contained in a license demand from a license storage means, A processing means to add terminal identification information to the license extracted by the extract means, It is characterized by having a signature means to add electronic signature to the license to which terminal identification information was

added by the processing means, and a transmitting means to transmit the license signed by the signature means to the information processor which transmitted the license demand, using the private key of a license server.

[0018] Contents specific information for the information processing approach of this invention to specify the contents by which utilization authorization is carried out, The step which memorizes the license including the terminal identification information which identifies an information processor, The step which receives the license demand containing the license identification information which was transmitted from the information processor, and which identifies a license, The step which extracts the license corresponding to the license identification information contained in a license demand from a license storage means, The step which adds terminal identification information to the license extracted by the extract means, It is characterized by including the step which adds electronic signature to the license to which terminal identification information was added by the processing means, and the step which transmits the license signed by the signature means to the information processor which transmitted the license demand using the private key of a license server.

[0019] In a program, contents are decoded a condition [holding the license] in the information processor of this invention, the information processing approach, and a list, and it is made available.

[0020] By the information processing approach, an effective license is published only with a specific information processor in the license server of this invention, and a list.

[0021]

[Embodiment of the Invention] Drawing 1 shows the contents offer structure of a system which applied this invention. A client 1-1 and 1-2 (hereafter, when these clients do not need to be distinguished separately, a client 1 is only called) are connected to the Internet 2. In this example, although two clients are shown, the client of the number of arbitration is connected to the Internet 2.

[0022] Moreover, when the contents server 3 which offers contents to a client 1, the license server 4 which gives a license required to use the contents which the contents server 3 offers to a client 1, and a client 1 receive a license, the accounting server 5 which performs accounting to the client 1 is connected to the Internet 2.

[0023] These contents servers 3, a license server 4, and the accounting server 5 are also connected to the number of arbitration, and the Internet 2.

[0024] Drawing 2 expresses the configuration of a client 1.

[0025] In drawing 2 , CPU (Central Processing Unit)21 performs various kinds of processings according to the program memorized by ROM (Read Only Memory)22 or

the program loaded to RAM (Random Access Memory)23 from the storage section 28. a timer 20 -- a time check -- it operates and time information is supplied to CPU21. To RAM23, CPU21 performs various kinds of processings upwards again, and required data etc. are memorized suitably.

[0026] The encryption decode section 24 performs processing which decodes the already enciphered contents data while enciphering contents data. Contents data are encoded by ATRAC(Adaptive Transform Acoustic Coding)3 method etc., and the codec section 25 is made to supply and record on the semiconductor memory 44 connected to the drive 30 through the input/output interface 32. Or the codec section 25 decodes the data which were read from semiconductor memory 44 through the drive 30 and which are encoded again.

[0027] Semiconductor memory 44 is constituted by the memory stick (trademark) etc.

[0028] CPU21, ROM22, RAM23, the encryption decode section 24, and the codec section 25 are mutually connected through the bus 31. The input/output interface 32 is also connected to this bus 31 again.

[0029] The communications department 29 which consists of the storage section 28 which consists of a display which consists of the input section 26 which consists of a keyboard, a mouse, etc., CRT, LCD, etc., the output section 27 which becomes a list from a loudspeaker etc., a hard disk, etc., a modem, a terminal adopter, etc. is connected to the input/output interface 32. The communications department 29 performs the communications processing through the Internet 2. The communications department 29 performs the communications processing of an analog signal or a digital signal among other clients again.

[0030] Drive 30 is connected to an input/output interface 32 again if needed, it is suitably equipped with a magnetic disk 41, an optical disk 42, a magneto-optic disk 43, or semiconductor memory 44, and the computer program by which reading appearance was carried out from them is installed in the storage section 28 if needed.

[0031] In addition, although a graphic display is omitted, the contents server 3, a license server 4, and the accounting server 5 are also constituted by the client 1 shown in drawing 2 , and the computer which has the same configuration fundamentally. Then, in the following explanation, the configuration of drawing 2 is quoted also as a configuration of the contents server 3, a license server 4, the accounting server 5, etc.

[0032] Next, with reference to the flow chart of drawing 3 , the processing whose client 1 receives offer of contents from the contents server 3 is explained.

[0033] When a user orders it access to the contents server 3 by operating the input

section 26, CPU21 controls the communications department 29 and he makes it access the contents server 3 through the Internet 2 in step S1. In step S2, a user operates the input section 26, and if the contents which receive offer are specified, CPU21 will notify the contents specified as the contents server 3 through the Internet 2 from reception and the communications department 29 in this assignment information. Since the contents data with which the carrier beam contents server 3 was enciphered in this advice are transmitted so that it may mention later with reference to the flow chart of drawing 4 , in step S3, CPU21 will supply and store that contents data enciphered in the hard disk which constitutes the storage section 28 in step S4, if this contents data is received through the communications department 29.

[0034] Next, with reference to the flow chart of drawing 4 , contents offer processing of the contents server 3 corresponding to the above processing of a client 1 is explained. In addition, in the following explanation, the configuration of the client 1 of drawing 2 is quoted also as a configuration of the contents server 3.

[0035] In step S21, when it stands by until CPU21 of the contents server 3 received access from the Internet 2 from the client 1 through the communications department 29, and access is judged to be a carrier beam, it progresses to step S22 and incorporates the information which specifies the contents transmitted from the client 1. The information which specifies these contents is information which the client 1 has notified in step S2 of drawing 3 .

[0036] In step S23, CPU21 of the contents server 3 reads the contents specified for the information incorporated by processing of step S22 out of the contents data memorized by the storage section 28. CPU21 supplies the contents data by which reading appearance was carried out from the storage section 28 to the encryption decode section 24, and makes them encipher in step S24 using the contents key Kc.

[0037] Since the contents data memorized by the storage section 28 are already encoded by the codec section 25 with ATRAC3 method, this contents data encoded will be enciphered.

[0038] In addition, of course, the storage section 28 can be made to memorize contents data in the condition of having enciphered beforehand. In this case, processing of step S24 can be omitted.

[0039] Next, in step S25, CPU21 of the contents server 3 adds the license ID for discriminating a license required using contents from key information (EKB and KEKBC (Kc) which are later mentioned with reference to drawing 5) required decoding the contents enciphered to the header which constitutes the format which transmits the enciphered contents data. And in step S26, CPU21 of the contents

server 3 transmits the data which formatted the key and the header which added License ID to the accessed client 1 through the Internet 2 from the communications department 29 at the contents enciphered by processing of step S24, and processing of step S25.

[0040] Drawing 5 is carried out in this way, and expresses the configuration of a format in case contents are supplied to a client 1 from the contents server 3. This format is constituted by a header (Header) and data (Data) as shown in this drawing.

[0041] The data KEKBC (Kc) as contents information (Content information), digital-rights management information (DRM(Digital Right Management) information), License ID (License ID), an INEBU ring key block (validation key block) (EKB (EnablingKey Block)), and a contents key Kc enciphered using the key KEKBC generated from EKB are arranged at the header. In addition, about EKB, it mentions later with reference to drawing 15 .

[0042] Information, such as a method of the content ID (CID) as identification information for identifying the contents data by which formatting is carried out as data, and the codec of the contents, is included in contents information.

[0043] The regulation and condition (Usage rules/status) which use contents, and URL (Uniform Resource Locator) are arranged at digital-rights management information. The count of playback of contents, the count of a copy, etc. are described by an activity regulation and the condition.

[0044] URL is address information accessed when acquiring the license specified with License ID, and is the address of the license server 4 specifically required in the case of the system of drawing 1 , since it is licensed. License ID identifies the license needed when using the contents currently recorded as data.

[0045] Data are constituted by the encryption block (Encryption Block) of the number of arbitration. Each encryption block is constituted by an initial vector (IV (Initial Vector)), seed (Seed), and data EK'c (data) that enciphered contents data by key K'c.

[0046] Key K'c is constituted by the value calculated to the Hash Function with the application of the contents key Kc and the value Seed set up by random numbers as shown in a degree type.

[0047] K'c=Hash (Kc, Seed) [0048] The initial vector IV and Seed Seed are set as a different value for every encryption block.

[0049] This encryption classifies the data of contents per 8 bytes, and is performed every 8 bytes. 8 bytes of latter encryption is performed in the CBC (Cipher Block Chaining) mode performed using the result of 8 bytes of encryption of the preceding paragraph.

[0050] Since 8 bytes of encryption result of the preceding paragraph does not exist when enciphering 8 bytes of first contents data in the case of the CBC mode, when enciphering 8 bytes of first contents data, encryption is performed by making the initial vector IV into initial value.

[0051] By performing encryption by this CBC mode, even if one encryption block is decoded, it is controlled that that effect attains to other encryption blocks.

[0052] In addition, about this encryption, drawing 46 is made reference and explained in full detail behind.

[0053] Moreover, about a cipher system, it does not restrict to this.

[0054] A client 1 is no charge about the contents server 3 to contents as mentioned above, and can be acquired freely. Therefore, the contents itself become possible [distributing to a large quantity].

[0055] However, each client 1 needs to hold the license, when using the acquired contents. Then, with reference to drawing 6 , processing in case a client 1 reproduces contents is explained.

[0056] In step S41, CPU21 of a client 1 acquires the identification information (CID) of the contents directed because a user operates the input section 26. This identification information is constituted by the title of contents, the number given for each [which is memorized] contents of every.

[0057] And CPU21 will read the license ID corresponding to the contents (ID of a license required to use the contents), if contents are directed. This license ID is described by the header of the contents data enciphered as shown in drawing 5 .

[0058] Next, it progresses to step S42, the license corresponding to the license ID in which CPU21 was read at step S41 is already acquired by the client 1, and it judges whether the storage section 28 memorizes. When the license is not acquired, it progresses to step S43 and, as for CPU21, license acquisition processing is still performed. The detail of this license acquisition processing is later mentioned with reference to the flow chart of drawing 7 .

[0059] In step S42, when judged with the license already being acquired, or when [as a result of performing license acquisition processing] a license is acquired in step S43, it progresses to step S44 and CPU21 judges whether the license acquired is a thing within an expiration date. It is judged to be the length (to refer to drawing 8 mentioned later) specified as a content of the license by comparing with the time of present in Japan [which is clocked by the timer 20] whether a license is a thing within an expiration date. When judged with the expiration date of a license having already expired, CPU21 progresses to step S45, and performs a license update process. The

detail of this license update process is later mentioned with reference to the flow chart of drawing 10 .

[0060] When judged with a license being still within an expiration date, or when a license is updated in step S45, it progresses to step S46, and CPU21 reads the contents data enciphered from the storage section 28, and is made to store them in RAM23 in step S44. And it is the encryption block unit arranged at the data of drawing 5 , the data of the encryption block memorized by RAM23 are supplied to the encryption decode section 24, and CPU21 makes them decode in step S47 using the contents key Kc.

[0061] Although the example of the approach of obtaining the contents key Kc is later mentioned with reference to drawing 15 , it can obtain the key KEKBC contained in EKB (drawing 5) using a device node key (DNK) (drawing 8), and can obtain the contents key Kc from Data KEKBC (Kc) and (drawing 5) using the key KEKBC.

[0062] CPU21 makes the codec section 25 supply and decode further the contents data decoded by the encryption decode section 24 in step S48. And D/A conversion of the data decoded by the codec section 25 is supplied and carried out to the output section 27 from an input/output interface 32, and CPU21 makes them output from a loudspeaker.

[0063] Next, with reference to the flow chart of drawing 7 , the detail of the license acquisition processing performed at step S43 of drawing 6 is explained.

[0064] The client 1 acquires the service data containing the pair of Leaves ID and DNK (Device Node Key), and the private key and public key of a client 1, the public key of a license server, and the certificate of each public key by registering with a license server in advance.

[0065] It is a device node key required for Leaf ID to express the identification information assigned for every client, and for DNK decode the contents key Kc which is contained in EKB (validation key block) corresponding to the license and which is enciphered (with reference to drawing 12 , it mentions later).

[0066] In step S61, CPU21 acquires first URL corresponding to the license ID now made into the processing object from the header shown in drawing 5 . As mentioned above, this URL is the address which should be accessed when acquiring the license corresponding to the license ID too described by the header. Then, in step S62, CPU21 accesses URL acquired at step S61. Specifically, access is performed to a license server 4 by the communications department 29 through the Internet 2. At this time, a license server 4 requires the input of user ID and a password of the license assignment information and the list which specify the license (license required to use

contents) to purchase from a client 1 (step S102 of drawing 9 mentioned later). CPU21 displays this demand on the display of the output section 27. Based on this display, a user operates the input section 26 and enters license assignment information, user ID, and a password. In addition, the user of a client 1 accesses a license server 4 through the Internet 2, and acquires this user ID and password in advance.

[0067] In steps S63 and S64, CPU21 incorporates user ID and a password while incorporating the license identification information inputted from the input section 26. CPU21 makes the license demand which controls the communications department 29 and contains the inputted user ID and the leaf ID contained in license assignment information and service data (it mentions later) in a password transmit to a license server 4 through the Internet 2 in step S65.

[0068] a license server 4 is later mentioned with reference to drawing 9 -- as -- user ID, a password, and a list -- license assignment information -- being based -- a license -- transmitting (step S109) -- or a license is not transmitted when conditions are not fulfilled (step S112).

[0069] When it judges whether the license has been transmitted from the license server 4 and the license has been transmitted, CPU21 progresses to step S67, and makes the storage section 28 supply and memorize the license in step S66.

[0070] In step S66, when it judges with a license not being transmitted, CPU21 progresses to step S68, and performs error processing. Since the license for using contents is not acquired, specifically, CPU21 forbids regeneration of contents.

[0071] It becomes possible to use the contents only after acquiring the license corresponding to the license ID with which each client 1 accompanies contents data as mentioned above.

[0072] In addition, license acquisition processing of drawing 7 can also be beforehand carried out, before each user acquires contents.

[0073] The license with which a client 1 is provided contains a service condition and leaf ID ****, as shown in drawing 8 .

[0074] The expiration date which can use contents for a service condition based on the license, The download length which can download contents based on the license, The count which can copy contents based on the license (count of a copy allowed), It is based on the count of check-out, the count of the maximum check-out, and its license. The information which shows the count which can copy contents to access recordable on CD-R and PD (Portable Device), the access which can shift a license to ownership (acquisition condition), the duty in which an activity log is taken, etc. is

included.

[0075] Next, with reference to the flow chart of drawing 9 , license offer processing of the license server 4 performed corresponding to license acquisition processing of the client 1 of drawing 7 is explained. In addition, the configuration of the client 1 of drawing 2 is quoted as a configuration of a license server 4 also in this case.

[0076] In step S101, CPU21 of a license server 4 stands by until it receives access from a client 1, and it requires transmission of license assignment information of user ID, a password, and a list from the client 1 which progressed to step S102 at the time of a carrier beam, and has accessed access. As it mentioned above, when license assignment information (license ID) has been transmitted to user ID, the password, and the leaf ID list by processing of step S65 of drawing 7 from the client 1, CPU21 of a license server 4 performs processing which receives and incorporates this through the communications department 29.

[0077] and the credit of the user corresponding to [CPU21 of a license server 4 accesses the accounting server 5 from the communications department 29 in step S103, and] user ID and a password -- processing is required. the accounting server 5 -- the Internet 2 -- minding -- the credit from a license server 4 -- the credit which permits grant of a license when the demand of processing is received, the payment hysteresis of the past of the user corresponding to the user ID and password etc. is investigated, it investigates whether there is any track record of the nonpayment of the countervalue of a license of the user in the past and there is such no track record -- the case where a result is transmitted and there is a track record of nonpayment etc. -- the credit of the disapproval of licensing -- a result is transmitted.

[0078] step S104 -- setting -- CPU21 of a license server 4 -- the credit from the accounting server 5 -- the credit which permits that a result gives a license -- when it judges whether it is a result and grant of a license is permitted, it progresses to step S105 and the license corresponding to the license assignment information incorporated by processing of step S102 takes out out of the license memorized by the storage section 28. As for the license memorized by the storage section 28, information, such as License ID, a version, the date and time of creation, and an expiration date, is described beforehand. In step S106, CPU21 adds the leaf ID received with the license. Furthermore, in step S107, CPU21 chooses the service condition matched with the license chosen at step S105. Or by processing of step S102, when a service condition is specified from a user, it is added to the service condition for which the service condition is prepared beforehand if needed again. CPU21 adds the selected service condition to a license.

[0079] In step S108, CPU21 signs a license with the private key of a license server, and, thereby, the license of a configuration as shown in drawing 8 is generated.
[0080] Next, it progresses to step S109 and CPU21 of a license server 4 makes the license (it has the configuration shown in drawing 8) transmit to a client 1 through the Internet 2 from the communications department 29.
[0081] CPU21 of a license server 4 makes the storage section 28 memorize the license (for a service condition and Leaf ID to be included) which is processing of step S109 and was transmitted now in step S110 corresponding to the user ID and the password which were incorporated by processing of step S102. Furthermore, in step S111, CPU21 performs accounting. Specifically, CPU21 requires the accounting to the user corresponding to the user ID and password of the accounting server 5 from the communications department 29. The accounting server 5 performs accounting to that user based on the demand of this accounting. As mentioned above, when the payment [that user] to this accounting, it can be licensed even if that user demands grant of a license henceforth.
[0082] namely, the credit which makes grant of a license disapproval from the accounting server 5 in this case -- since a result is transmitted, it progresses to step S112 from step S104, and CPU21 performs error processing. CPU21 of a license server 4 outputs the message of the purport which cannot give a license to the client 1 which controlled the communications department 29 and has accessed it, and, specifically, terminates processing.
[0083] In this case, since that client 1 cannot be licensed as mentioned above, using those contents (decoding a code) can be performed.
[0084] Drawing 10 expresses the detail of the license update process in step S45 of drawing 6 . Processing of step S131 of drawing 10 thru/or step S135 is the same processing as processing and the basic target of step S61 of drawing 7 thru/or step S65. However, in step S133, CPU21 incorporates the license ID of the license instead of the license to purchase to update. And in step S135, CPU21 transmits the license ID of the license updated with user ID and a password to a license server 4.
[0085] Corresponding to transmitting processing of step S135, a license server 4 presents a service condition so that it may mention later (step S153 of drawing 11). Then, in step S136, CPU21 of a client 1 receives presentation of the service condition from a license server 4, and outputs and displays this on the output section 27. A user operates the input section 26, and a predetermined service condition is chosen out of this service condition, or he newly adds a predetermined service condition. CPU21 transmits the application for purchasing the service condition (conditions which

update a license) chosen as mentioned above to a license server 4 at step S137. Corresponding to this application, a license server 4 transmits a final service condition so that it may mention later (step S154 of drawing 11). Then, in step S138, CPU21 of a client 1 acquires the service condition from a license server 4, and updates the service condition as a service condition of the corresponding license already memorized by the storage section 28 in step S139.

[0086] Drawing 11 expresses the license update process which a license server 4 performs corresponding to the license update process of the above client 1.

[0087] First, in step S151, CPU21 of a license server 4 will receive the license assignment information which the client 1 transmitted at step S135 with the renewal demand information of a license in step S152, if access from a client 1 is received.

[0088] In step S153, if an updating demand of a license is received, CPU21 will read the service condition (service condition to update) corresponding to the license from the storage section 28, and will transmit it to a client 1.

[0089] If it applies for the purchase of a service condition from a client 1 by processing of step S137 of drawing 10 to this presentation as mentioned above, in step S154, CPU21 of a license server 4 will generate the data corresponding to the service condition for which it applied, and will transmit them to a client and 1 in step S154. A client 1 updates the service condition of the already registered license using the service condition received by processing of step S139, as mentioned above.

[0090] In this invention, as shown in drawing 12 , the key of a device and a license is managed based on the principle of a broadcasting in chestnut PUSHON (Broadcast Encryption) method. A key is made into a hierarchy tree structure and the leaf (leaf) of the bottom corresponds to the key of each device. In the case of the example of drawing 12 , 16 devices from a number 0 to a number 15 or the key corresponding to a license is generated.

[0091] Each key is specified corresponding to each node of the tree structure shown by the drawing Nakamaru mark. In this example, the key K000 thru/or the key K111 correspond [corresponding to the root node of the maximum upper case / the root key KR / keys K0 and K1 / a key K00 thru/or K11] corresponding to the node of the 4th step corresponding to the 3rd step of node, respectively corresponding to the 2nd step of node. And a key K0000 thru/or K1111 support the leaf (device node) as a node of the bottom, respectively.

[0092] Since it considers as the layered structure, the key of the high order of a key K0010 and a key 0011 is set to K001, and the key of the high order of a key K000 and a key K001 is set to K00. Like the following, the key of the high order of a key K00 and

a key K01 is set to K0, and the key of the high order of a key K0 and a key K1 is set to KR.

[0093] The key using contents is managed by the key corresponding to each node of one pass from the device node (leaf) of the bottom to the root node of the maximum upper case. For example, based on the license corresponding to the node (leaf ID) of a number 3, the key using contents is managed by each key of the pass containing Keys K0011, K001, K00, K0, and KR.

[0094] In the system of this invention, as shown in drawing 13 , it is the keying system constituted based on the principle of drawing 12 , and management of the key of a device and the key of a license is performed. In the example of drawing 13 , 8+24+32 steps of nodes are made into a tree structure, and a category corresponds to each node from a root node to eight steps of low order. The category in here means categories, such as a category of the device which uses semiconductor memory, such as a memory stick, and a category of a device which receives digital broadcasting. And this system (T system is called) corresponds to one node in this category node as a system which manages a license.

[0095] That is, a license corresponds by the key corresponding to 24 steps of a younger hierarchy's nodes further from the node of this T system. In the case of this example, thereby, the license of 224 (about 16 mega) can be specified. Furthermore, 32 steps of lower hierarchies can prescribe the user (or client 1) of 232 (about 4giga). The key corresponding to 32 steps of nodes of the bottom constitutes DNK (Device Node Key), and let ID corresponding to the leaf of the bottom be Leaf ID.

[0096] Each device and the key of a license correspond to one of the pass which consists of each node of 64 (= 8+24+32) stages. For example, the contents key which enciphered contents is enciphered using the key corresponding to the node which constitutes the pass assigned to the corresponding license. It is enciphered using the key of the hierarchy of the latest low order, and the key of the hierarchy of a high order is arranged in EKB (with reference to drawing 15 , it mentions later). DNK of the bottom is not arranged in EKB, but is described by service data, and is given to a user's client 1. A client 1 decodes the key of the hierarchy on it to the pan described in EKB using the key which decoded the key of the hierarchy of the latest high order described in EKB (drawing 15) distributed with contents data, and decoded and obtained it using DNK described by the license. By performing the above processing one by one, a client 1 can obtain all the keys belonging to the pass of the license.

[0097] The concrete example of the classification of the category of a hierarchy tree structure to drawing 14 is shown. In drawing 14 , the root key KR2301 is set to the

maximum upper case of a hierarchy tree structure, the node key 2302 is set to the following intermediate stages, and the leaf key 2303 is set to the bottom. Each device holds each leaf key, and a series of node keys and root key from a leaf key to a root key.

[0098] The predetermined node of the Mth step (the example of drawing 13 M= 8) is set up as a category node 2304 from the maximum upper case. That is, let each of the node of the Mth step be the device setting-out node of a specific category. Let M+1 or less step of node, and a leaf be the node and leaf about the device contained in the category by making one node of the Mth step into top-most vertices.

[0099] For example, a category [a memory stick (trademark)] is set to one node 2305 of the Mth step of drawing 14 , and the node which stands in a row below in this node, and a leaf are set up as the node or leaf containing various devices which used memorandum RISUTEIIKU only for categories. That is, 2305 or less node is defined as the related node of the device defined as the category of a memory stick, and a set of a leaf.

[0100] Furthermore, a low-ranking stage can be set up as a subcategory node 2306 by several steps from M steps. In the example of drawing 14 , the node 2306 of [the vessel only for playbacks] is set up as a subcategory node contained in the category of the device which used the memory stick for the node under two steps of the category [memory stick] node 2305. furthermore -- sub -- a category -- a node -- it is -- playback -- dedication -- a vessel -- a node -- 2306 -- less than -- playback -- dedication -- a vessel -- a category -- containing -- having -- music -- a regenerative function -- with -- a telephone -- a node -- 2307 -- setting up -- having -- further -- the -- low order -- music -- a regenerative function -- with -- a telephone -- a category -- containing -- having -- [-- PHS --] -- a node -- 2308 -- [-- a cellular phone --] -- a node -- 2309 -- setting up -- having -- **** .

[0101] Furthermore, a category and a subcategory can be set up in units (these are generically called an entity hereafter) of arbitration, such as not only the class of device but the node which a certain manufacturer, a content provider, a settlement-of-accounts engine, etc. manage uniquely, for example, i.e., a batch, a jurisdiction unit, or an offer service unit. For example, if it sets up as a top-most-vertices node only for game device XYZ(s) to which a game equipment manufacturer sells one category node To the game device XYZ which a manufacturer sells, the node key of the lower berth below the top-most-vertices node, It becomes possible to store and sell a leaf key. Distribution of after that and encryption contents, Or the validation key block (EKB) constituted by the node key below the

top-most-vertices node key and the leaf key in distribution of various keys and an update process is generated and distributed, and distribution of available data is attained only to the device below a top-most-vertices node.

[0102] Thus, by considering as the configuration which sets up the following nodes as a related node of the category defined as the top-most-vertices node, or a subcategory by making one node into top-most vertices The validation key block (EKB) to which the manufacturer who manages one top-most-vertices node of a category stage or a subcategory stage, a content provider, etc. make the node top-most vertices is generated uniquely. The configuration distributed to the device belonging to below a top-most-vertices node is attained, and renewal of a key can be performed, without affecting at all the device belonging to the node of other categories which do not belong to a top-most-vertices node.

[0103] For example, in the tree structure shown in drawing 12 , four devices 0, 1, 2, and 3 contained in one group hold the keys K00, K0, and KR common as a node key. By using this node key share configuration, it becomes possible to provide only devices 0, 1, 2, and 3 with a common contents key. For example, if node key K00 the very thing held in common is set up as a contents key, setting out of a contents key only with common devices 0, 1, 2, and 3 is possible, without performing new key sending. Moreover, if the value Enc (K00, Kcon) which enciphered the new contents key Kcon by the node key K00 is stored in a record medium through a network and distributed to devices 0, 1, 2, and 3 Only devices 0, 1, 2, and 3 become possible [solving Code Enc (K00 Kcon) using the share node key K00 held in each device, and obtaining the contents key Kcon]. In addition, it is shown that Enc (Ka, Kb) is data which enciphered Kb by Ka.

[0104] Moreover, when the keys K0011, K001, K00, K0, and KR which a device 3 owns having been analyzed by the aggressor (hacker), and having exposed is revealed, in order to protect the data transmitted and received by the system (group of devices 0, 1, 2, and 3) after it in t at a certain event, it is necessary to separate a device 3 from a system. For that purpose, it is necessary to update the node keys K001, K00, K0, and KR to respectively new key K (t)001 and K (t)00 and K (t)0 and K(t) R, and to tell the updating key to devices 0, 1, and 2. Here, it is shown that K(t) aaa is the updating key of the generation (Generation) t of Key Kaaa.

[0105] distribution **** of an updating key -- it ****** just. Renewal of a key is performed by storing the table constituted by the block data called the validation key block (EKB:Enabling Key Block) shown for example, in drawing 15 A in a record medium, and supplying it to devices 0, 1, and 2 through a network. In addition, a

validation key block (EKB) is constituted by the cryptographic key for distributing the key newly updated by the device corresponding to each leaf (node of the bottom) which constitutes a tree structure as shown in drawing 12 . A validation key block (EKB) may be called the renewal block of a key (KRB:Key Renewal Block).

[0106] The validation key block (EKB) shown in drawing 15 A is constituted as block data with the data configuration which can update only the required device of renewal of a node key. The example of drawing 15 A is the block data formed for the purpose of distributing Generation's t updating node key in the devices 0, 1, and 2 in the tree structure shown in drawing 12 . K (t)00 and K (t)0 and K(t) R are required for a device 0 and a device 1 as an updating node key, and K (t)001 and K (t)00 and K (t)0 and its K(t) R are required for a device 2 as an updating node key so that clearly from drawing 12 .

[0107] As shown in EKB of drawing 15 A, two or more cryptographic keys are contained in EKB. The cryptographic key of the bottom of drawing 15 A is Enc (K0010, K(t)001). this -- a device -- two -- having -- a leaf -- a key -- K -- 0010 -- enciphering -- having had -- updating -- a node -- a key -- K -- (-- t --) -- 001 -- it is -- a device -- two -- self -- having -- a leaf -- a key -- K -- 0010 -- this cryptographic key -- decoding -- the updating node key K -- 001 can be obtained. moreover, the updating node key K obtained by decode -- 001 -- using -- decode of the 2nd step of cryptographic key Enc (K (t)001 and K (t) -- 00) is possible from under drawing 15 A -- becoming -- the updating node key K -- 00 can be obtained.

[0108] decoding the 2nd step of cryptographic key Enc (K (t)00 and K (t) -- 0) from on drawing 15 A one by one below -- the updating node key K -- 0 is obtained and updating root key K(t) R is obtained from on drawing 15 A using this by decoding the 1st step of cryptographic key Enc (K (t)0, K(t) R).

[0109] On the other hand, the node key K000 is not contained in the object to update, but the things whose nodes 0 and 1 are need as an updating node key are K (t)00 and K (t)0 and K(t) R. Nodes 0 and 1 acquire updating node key K(t)00 using the debye skiings K0000 and K0001 by decoding the 3rd step of cryptographic key Enc (K000, K(t)00) from on drawing 15 A. one by one below by decoding the 2nd step of cryptographic key Enc (K (t)00 and K (t) -- 0) from on drawing 15 A Updating node key K (t) 0 is obtained and updating root key K(t) R is obtained from on drawing 15 A by decoding the 1st step of cryptographic key Enc (K (t)0, K(t) R). Thus, devices 0, 1, and 2 can obtain updated key K(t) R.

[0110] In addition, the index of drawing 15 A shows the absolute address of the node key used as a decode key for decoding the cryptographic key on the right-hand side

of drawing, and a leaf key.

[0111] It is unnecessary, and renewal of node key [of the high order stage of a tree structure shown in drawing 12] K (t)0 and K(t) R can distribute updating node key K(t)00 to devices 0, 1, and 2 by using the validation key block (EKB) of drawing 15 B, when only the node key K00 needs to be updated.

[0112] EKB shown in drawing 15 B is available when distributing the new contents key shared in a specific group. As an example, the record medium with the devices 0, 1, 2, and 3 in the group who shows by the dotted line is used for drawing 12 , and new common contents key K(t) con presupposes that it is required. this -- the time -- a device -- zero -- one -- two -- three -- being common -- a node -- a key -- K -- 00 -- having updated -- K -- (-- t --) -- 00 -- using -- being new -- being common -- updating -- contents -- a key -- K -- (-- t --) -- con -- having enciphered -- data -- Enc (K (t) 00 K(t) con) -- drawing 15 -- B -- being shown -- having -- EKB -- distributing -- having . By this distribution, the distribution of a device 4 etc. as data which cannot decode other groups' device is attained.

[0113] namely, -- a device -- zero -- one -- two -- EKB -- processing -- having obtained -- a key -- K -- (-- t --) -- 00 -- using -- a cipher -- decoding -- if -- t -- an event -- contents -- a key -- K -- (-- t --) -- con -- obtaining -- things -- possible -- becoming .

[0114] Processing of the device 0 which received the data Enc (K (t) 00 K(t) con) which enciphered new common contents key K(t) con to drawing 16 , using K(t)00 as an example of processing which obtains contents key K(t) con in t event, and EKB shown in drawing 15 B through the record medium is shown. That is, this example is an example which set the encryption message data based on EKB to contents key K(t) con.

[0115] As shown in drawing 16 , a device 0 generates node key K(t)00 by same EKB processing with having mentioned above using EKB at the generation t event stored in the record medium, and the node key K000 which he stores beforehand. furthermore, the updating node key K which decoded the device 0 -- using 00, updating contents key K(t) con is decoded, and in order to use it behind, it enciphers and stores by the leaf key K0000 which he has.

[0116] The example of a format of a validation key block (EKB) is shown in drawing 17 . A version 601 is an identifier which shows the version of a validation key block (EKB). In addition, a version has the function to identify the newest EKB, and the function which shows response relation with contents. A depth shows the number of hierarchies of the hierarchy tree to the device of the distribution place of a validation

key block (EKB). A data pointer 603 is a pointer in which the location of the data division 606 in a validation key block (EKB) is shown, and the tag pointer 604 is a pointer which the location of the tag section 607 and the signature pointer 605 show the location of signature 608.

[0117] Data division 606 store the data which enciphered the node key updated, for example. For example, each cryptographic key about the updated node key as shown in drawing 16 etc. is stored.

[0118] The tag section 607 is a tag in which the physical relationship of the enciphered node key which was stored in data division 606 and a leaf key is shown. The grant rule of this tag is explained using drawing 18 .

[0119] Drawing 18 shows the example which sends the validation key block (EKB) previously explained by drawing 15 A as data. The data at this time come to be shown in the table of drawing 18 B. Let the address of the top node contained in the cryptographic key at this time be a top node address. Since updating key K(t) R of a root key is contained in the case of this example, a top node address serves as KR. The data Enc (K (t)0, K(t) R) at this time, for example, the maximum upper case, correspond to the location P0 shown in the hierarchy tree shown in drawing 18 A. the data of the next stage are Enc (K (t)00 and K (t) -- 0), and correspond to the location P00 at the lower left of front data on a tree. When it sees from the position of a tree structure and data are in the bottom of it, a tag is set as 0 and a tag is set as 1, when there is nothing. A tag is set up as {a left (L) tag and a right (R) tag}. Since there are data in the location POO at the lower left of the location PO corresponding to the data Enc (K (t)0, K(t) R) of the maximum upper case of drawing 18 B and there are no data in L tag =0 and the right, it is set to R tag =1. Hereafter, a tag is set as all data and the data stream shown in drawing 18 C and a tag train are constituted.

[0120] A tag is set up in order that the corresponding data Enc (Kxxx, Kyyy) may show where [of a tree structure] it is located. the key data Enc (Kxxx, Kyyy) stored in data division 606 -- although it is only enumeration data of a key with which ... was enciphered simply, distinction of the location on the tree of the cryptographic key stored as data with the tag mentioned above is attained. The node index to which encryption data were made to correspond is used like a configuration of that previous drawing 15 explained, without using the tag mentioned above, for example, it is 0:Enc (K (t)0, K(t) R).

00:Enc(K(t)00,K(t)0

000:Enc(K((t)000,K(t)00)

Although considering as a data configuration like ... is also possible, if it is a

configuration using such an index, in the distribution which it becomes redundant data, and the amount of data increases, and minds a network, it is not desirable. On the other hand, distinction of a key position is attained by the small amount of data by using the tag mentioned above as index data in which a key position is shown.

[0121] It returns to drawing 17 and an EKB format is explained further. Signature (Signature) 608 is electronic signature which published the validation key block (EKB) and which a key management center (license server 4), contents ROBAIDA (contents server 3), a settlement-of-accounts engine (accounting server 5), etc. perform, for example. It checks that the device which received EKB is the validation key block (EKB) which the just validation key block (EKB) publisher published by signature verification.

[0122] When processing using the contents supplied from the contents server 3 is summarized based on the license supplied from the license server 4 as mentioned above, it comes to be shown in drawing 19 .

[0123] That is, while contents are offered from the contents server 3 to a client 1, a license is supplied to a client 1 from a license server 4. Contents are enciphered by the contents key Kc (Enc (Kc, Content)), the contents key Kc is added to the contents which were enciphered by the root key KR (it is the key obtained from EKB and corresponds to the key KEKBC in drawing 5) (Enc (KR, Kc)), and were enciphered with EKB, and a client 1 is provided with it.

[0124] As shown in drawing 20 , the root key KR enciphered by DNK is contained in EKB in the example of drawing 19 (Enc (DNK, KR)). Therefore, a client 1 can obtain the root key KR from EKB using DNK contained in service data. Furthermore, using the root key KR, the contents key Kc can be decoded from Enc (KR, Kc), and contents can be decoded from Enc (Kc, Content) using the contents key Kc.

[0125] Thus, according to the principle explained with reference to drawing 12 and drawing 15 , RIBOKU (revoke) of each client 1 becomes possible by assigning DNK to a client 1 according to an individual.

[0126] Moreover, by adding and distributing the license leaf ID, in a client 1, matching of service data and a license will be performed and it becomes possible to prevent the illegal copy of a license.

[0127] Moreover, it also enables an end user to create the contents which can prevent an illegal copy by distributing the certificate and private key for clients as service data using these.

[0128] About utilization of a certificate and a private key, it mentions later with reference to the flow chart of drawing 28 .

[0129] In this invention, since T system which manages a license, and the category using various kinds of contents of a device are matched with a category node as explained with reference to drawing 13 , two or more DNK(s) can be given to the same device. Consequently, it becomes possible to manage the contents of a different category with one device.

[0130] Drawing 21 expresses this relation. That is, based on T system, the license using contents 1 to which DNK1 is assigned is recorded on a device D1. Similarly, the contents 2 to which DNK2 was assigned and which carried out ripping to the memory stick from CD are recordable on this device D1. In this case, a device D1 becomes possible [treating simultaneously contents which are called contents 1 and contents 2 and which were distributed by different system (T system and device managerial system)]. Such a thing cannot be performed, when assigning new DNK, and DNK already assigned is deleted and it is made to make only one DNK correspond to a device.

[0131] Moreover, it becomes possible to classify the inside of the same category into small meetings, such as a genre of contents, a label, a dealer, and distribution service, and to manage it using a subcategory, by assigning the license category 1 and the license category 2 which are shown in each of three square shapes each of 32 lower hierarchies at drawing 22 . [in / drawing 13]

[0132] In the example of drawing 22 , the license category 1 belongs to the genre of jazz, and the license category 2 belongs to the genre of a lock, for example. License ID makes the contents 1 and contents 2 which are 1 correspond to the license category 1, and is distributed to the user 1 thru/or the user 3 at it, respectively. The contents 3 of license ID 2, contents 4, and contents 5 are contained, and the user 1 and the user 3 are provided with the license category 2, respectively.

[0133] Thus, in this invention, the key management which became independent for every category is attained.

[0134] Moreover, DNK is beforehand embedded neither to a device nor media, but by the license server 4, in case registration processing is performed, the system which can purchase the key by the user can be realized by making it download to each device or media.

[0135] After it is created, even if contents are carried out in what kind of usage, are concerned, they are not in the usage and it is desirable in all applications that it is usable. For example, also in a different contents distribution service or the domain in which service conditions differ, it is desirable that the same contents can be used. In this invention, for this reason, as mentioned above, the certificate (certificates) of a

private key and the public key corresponding to it is distributed to each user (client 1) from the license server 4 as a certificate authority. Using the private key, each user can create a signature (signature), can add to contents, and can guarantee Shinsei (integrity) of contents, and can aim at alteration prevention of contents.

[0136] The example of processing in this case is explained with reference to the flow chart of drawing 23 . Processing of drawing 23 explains the ripping processing a user makes [processing] the storage section 28 memorize [processing] the data reproduced from CD.

[0137] First, in step S171, CPU21 of a client 1 incorporates the playback data of CD inputted through the communications department 29 as record data. In step S172, CPU21 judges whether the watermark is contained in the record data incorporated by processing of step S171. This watermark is constituted by the copy management information (CCI) of a triplet, and the 1-bit trigger (Trigger), and is embedded in the data of contents. When a watermark is detected, CPU21 progresses to step S173, and performs processing which extracts the watermark. When a watermark does not exist, processing of step S173 is skipped.

[0138] Next, in step S174, CPU21 creates the data of the header recorded corresponding to contents. The data of this header are constituted by URL showing the access place for acquiring content ID, License ID, and a license, and the watermark.

[0139] Next, it progresses to step S175 and CPU21 creates the digital signature based on the data of the header created by processing of step S174 using its own private key. This private key is acquired from a license server 4 (step S67 of drawing 7).

[0140] CPU21 controls the encryption decode section 24 by step S176, and contents are made to encipher by the contents key. A contents key is simultaneously acquired, when contents are acquired (drawing 5 or drawing 19).

[0141] Next, CPU21 makes data record on the magneto-optic disk 43 constituted with a mini disc etc. in step S177 based on a file format.

[0142] In addition, when a record medium is a mini disc, CPU21 supplies contents to the codec section 25, for example, makes contents encode with ATRAC3 method in step S176. And the encoded data are further enciphered by the encryption decode section 24.

[0143] Drawing 24 expresses typically the condition that contents were recorded on the record medium as mentioned above. The watermark (WM) extracted from the contents (E (At3)) enciphered is recorded out of contents (header).

[0144] Drawing 25 expresses the more detailed configuration of the file format in the case of recording contents on a record medium. In this example Content ID (CID), License ID (LID) URL and the header containing a watermark (WM) are recorded, and also EKB, the data which enciphered the contents key Kc by the root key KR (Enc (KR, Kc)), A certificate (Cert), the digital signature generated based on the header (Sig (Header)), The data (Enc (Kc, Content)), the metadata (Meta Data), and the mark (Mark) which enciphered contents by the contents key Kc are recorded.

[0145] Although the watermark is embedded to the interior of contents, as shown in drawing 24 and drawing 25 , the interior of contents is making it arrange in a header independently, and it becomes possible [detecting the information currently embedded to contents as a watermark promptly and simply]. Therefore, it can judge promptly whether the contents can be copied.

[0146] In addition, metadata expresses data, such as a jacket, a photograph, and words. About a mark, it mentions later with reference to drawing 31 .

[0147] Drawing 26 expresses the example of the public key certificate as a certificate. A public key certificate is a certificate which the certificate authority (CA:Certificate Authority) in a public key cryptosystem publishes, a certificate authority adds information, such as an expiration date, and adds the digital signature by the certificate authority further, and is usually drawn up by self ID, public key, etc. which the user submitted to the certificate authority. In this invention, since a license server 4 (or contents server 3) also publishes a certificate, a private key, therefore a public key, a user can get this public key certificate by providing a license server 4 with user ID, a password, etc., and performing registration processing.

[0148] A certificate user's public key is contained in the algorithm which used the public key certificate in drawing 26 for the version number of a certificate, the serial number of the certificate which a license server 4 assigns to the user (user) of a certificate, and a digital signature and a parameter, the identifier of a certificate authority (license server 4), the expiration date of a certificate, a certificate user's ID (Node ID or Leaf ID), and a list as a message. Furthermore, the digital signature created by the license server 4 as a certificate authority is added to this message. This digital signature is data generated using the private key of a license server 4 based on the hash value generated with the application of the Hash Function to the message.

[0149] in the case of the example of drawing 12 , if Node ID or Leaf ID is a device 0, it will be set to "0000", if it comes out device 1, it will be set to "0001", and if it is a device 15, it will be set to "1111." Based on such ID, it is identified whether the device

(entity) is the entity located in which location (a leaf or node) of a tree configuration.
[0150] Thus, distribution of contents will be freely performed by dissociating and distributing a license required using contents with contents. The contents which came to hand in the approach of arbitration or the path can be dealt with unitary.
[0151] Moreover, of course, when distributing the contents of the format through the Internet with constituting a file format as shown in drawing 25 , when providing for an SDMI (Secure Digital Music Initiative) device, it becomes possible to manage the copyright of contents.
[0152] Furthermore, for example, as shown in drawing 27 , even if contents are offered through a record medium, even if it is provided through the Internet 2, the same processing enables it to check out to predetermined PD (Portable Device) as an SDMI (Secure Digital Music Initiative) device etc.
[0153] Next, with reference to the flow chart of drawing 28 , processing in case a client 1 checks out contents to other clients (for example, PD) is explained.
[0154] First, in step S191, CPU21 judges whether the digital signature is added to contents. When judged with the digital signature being added, it progresses to step S192, and CPU21 extracts a certificate and performs processing verified with the public key of a certificate authority (license server 4). That is, a client 1 acquires the public key corresponding to the private key of a license server 4 to the license server 4, and decodes the digital signature added to the public key certificate with the public key. As explained with reference to drawing 26 , the digital signature is generated based on the private key of a certificate authority (license server 4), and can be decoded using the public key of a license server 4. Furthermore, CPU21 calculates a hash value with the application of a Hash Function to the whole message of a certificate. And if CPU21 compares the calculated hash value with the hash value which decoded the digital signature and was obtained and both corresponds, a message will judge with it not being what was altered. When both are not in agreement, it will be said that this certificate is altered.
[0155] Then, in step S193, when judged with whether the certificate is altered and or not judging CPU21 and not being altered, it progresses to step S194 and processing which verifies a certificate by EKB is performed. This verification processing is performed by investigating whether EKB can be followed or not based on the leaf ID (drawing 26) contained in a certificate. This verification is explained with reference to drawing 29 and drawing 30 .
[0156] Now, as shown in drawing 29 , suppose that it is the device [RIBOKU / device / the device which has the leaf key K1001]. At this time, EKB which has data

(cryptographic key) as shown in drawing 30 , and a tag is distributed to each device (leaf). This EKB is EKB which updates keys KR, K1, K10, and K100, in order RIBOKU [the device "1001" in drawing 29].

[0157] All leaves other than a RIBOKU device "1001" can acquire updated root key K(t) R. That is, since the leaf which stands in a row in the low order of the node key K0 holds in a device the node key K0 which is not updated, it can acquire updating root key K(t) R by decoding a cryptographic key Enc (K0, K(t) R) by the key K0.

[0158] Moreover, the leaf not more than node key K11 can acquire updating node key K(t)1 using the node key K11 which is not updated by decoding Enc (K11, K(t)1) by the node key K11. Furthermore, it becomes possible by decoding Enc (K (t)1, K(t) R) by node key K(t)1 to acquire updating root key K(t) R. Also about the low order leaf of the node key K101, it is possible to acquire updating root key K(t) R similarly.

[0159] Furthermore, the device "1000" which has the leaf key [RIBOKU / key] K1000 can decode Enc (K1000, K(t)100) by the leaf key K1000 of self, can acquire node key K(t)100, using this, can carry out the sequential decode of the node key of a high order, and can acquire updating root key K(t) R further.

[0160] on the other hand -- RIBOKU -- having had -- a device -- "-- 1001 -- " -- self -- a leaf -- one -- a step -- a top -- updating -- a node -- a key -- K -- (-- t --) -- 100 -- EKB -- processing -- being unacquirable -- since -- after all -- updating root key K(t) R -- being unacquirable .

[0161] EKB which has the data shown in drawing 30 and a tag is distributed and stored in the just device [RIBOKU / device] (client 1) from the license server 4.

[0162] Then, each client can perform EKB trace processing using the tag. This EKB trace processing is processing which judges whether a key distribution tree can be followed from the root key of a high order.

[0163] For example, "1001" which is ID (leaf ID) of the leaf "1001" of drawing 29 is grasped as 4 bits of "1", "0", "0", and "1", and it is judged one by one from the most significant bit whether a tree can be followed according to a lower bit. In this judgment, if a bit is 1, it will go to right-hand side, and if it is 0, processing which goes to left-hand side will be performed.

[0164] Since the most significant bit of ID "1001" is 1, it goes to right-hand side from the root key KR of drawing 29 . It is judged with the tag (tag of a number 0) of the beginning of EKB being 0: {0, 0}, and being what has data on both branches. In this case, since it can go to right-hand side, it can arrive at the node key K1.

[0165] Next, it progresses to the node of the low order of the node key K1. Since the 2nd bit of ID "1001" is 0, it goes to left-hand side. The tag in which the tag of a

number 1 expresses the existence of the data of the low order of the left-hand side node key K0, and the existence of the data of the low order of the node key K1 is shown is a tag of a number 2. As shown in drawing 30 , this tag shall be 2: {0, 0}, and shall have data on both branches. Therefore, it can go to left-hand side and can arrive at the node key K10.

[0166] Furthermore, the 3rd bit of ID "1001" is 0 and goes to left-hand side. At this time, the tag (tag of a number 3) in which the existence of the data of the low order of K10 is shown is 3: {0, 0}, and is judged to be what has data on both branches. Then, it can go to left-hand side and can arrive at the node key K100.

[0167] Furthermore, the least significant bit of ID "1001" is 1, and goes to right-hand side. The tag which the tag of a number 4 corresponds to the node key K11, and expresses the sign of the data of the low order of K100 is a tag of a number 5. This tag is 5: {0, 1}. Therefore, data will not exist in right-hand side. consequently, arrive at a node "1001" -- it is judged with there being nothing and the device of ID "1001" being the device which cannot acquire the updating root key by EKB, i.e., a RIBOKU device.

[0168] On the other hand, for example, the device ID which has the leaf key K1000 is "1000", and like the case where it mentions above, if EKB trace processing based on the tag in EKB is performed, it can arrive at a node "1000." Therefore, it is judged with the device of ID "I000" being a just device.

[0169] It returns to drawing 28 , and when RIBOKU [CPU21 judges RIBOKU / the certificate / based on verification processing of step S194 at step S195 and / the certificate], it progresses to step S196 and processing which verifies a digital signature with the public key contained in a certificate is performed.

[0170] That is, as shown in drawing 26 , a certificate user's (contents implementer) public key is contained in the certificate, and the signature (Sig (Header)) shown in drawing 25 is verified using this public key. That is, by comparing the hash value which calculated the digital signature Sig (Header) with the application of the Hash Function to the data (hash value) decoded and obtained and Header shown in drawing 25 using this public key, if both are in agreement, it can check that Header is not altered. On the other hand, it will be said that Header is altered if both are not in agreement.

[0171] In step S197, if judge CPU21 and it is not altered [whether Header is altered and or not], it progresses to step S198 and verifies a watermark. In step S199, CPU21 judges whether he can check out or not as a result of verification of a watermark. When you can check out, it progresses to step S200 and CPU21 performs check-out. That is, contents are made to transmit and copy to the client 1 of a check-out place.

[0172] When judged with a digital signature not existing in step S191, it sets to step

S193. When judged with the certificate being altered, it sets to step S195. When are judged with the certificate having been unverifiable by EKB and it is judged with the header being altered in step S197 as a result of verification of a digital signature, Or in step S199, when judged with prohibition of check-out being described by the watermark, it progresses to step S201 and error processing is performed. That is, check-out is forbidden in this case.

[0173] Thus, it becomes possible by distributing a certificate and a private key to a user from a license server 4, and adding a digital signature to contents creation time to guarantee Shinsei of the implementer of contents. Thereby, the negotiation of inaccurate contents can be controlled.

[0174] Furthermore, a watermark is detected to contents creation time, by ****** which gives the information to a digital signature, the alteration of watermark information can be prevented and Shinsei of contents can be guaranteed.

[0175] Consequently, even if the contents created once are distributed with what kind of gestalt, they become possible [guaranteeing Shinsei of the original contents].

[0176] Furthermore, since contents do not have a service condition but the service condition is added to the license, it is changing the service condition within a license, and it becomes possible to change the service conditions of the contents related to it all at once.

[0177] Next, the usage of a mark is explained. In this invention, as mentioned above, a service condition is added to the license instead of contents. However, an operating condition may change with contents. Then, in this invention, as shown in drawing 25 , a mark is added to contents.

[0178] Since a license and contents have the relation of one-pair **, it becomes difficult to describe each operating condition of contents only in the service condition of a license. Then, though management with a license is carried out by adding an operating condition to contents in this way, it becomes possible to manage each contents.

[0179] As shown in drawing 31 , a user's ID (leaf ID), an ownership flag, beginning-of-using time of day, a count of a copy, etc. are described by this mark.

[0180] Furthermore, the digital signature generated based on messages, such as Leaf ID, an ownership flag, beginning-of-using time of day, and a count of a copy, is added to a mark.

[0181] An ownership flag is added when the license only whose predetermined period makes contents usable is bought as it was (when duration of service is changed eternally). Beginning-of-using time of day is described when the activity of contents

is started within a predetermined period. For example, when the stage to download contents is restricted and download is performed within the length, the time which downloaded contents actually is described here. Thereby, it is proved that it is an effective activity within a period.

[0182] The count which resembled the count of a copy till then and copied the contents to it is described as hysteresis (log).

[0183] Next, when a user buys a license with reference to the flow chart of drawing 32 , the processing which adds a mark is explained as an example which adds a mark to contents.

[0184] First, in step S221, CPU21 accesses a license server 4 through the Internet 2 based on a command of the user from the input section 26.

[0185] In step S222, CPU21 incorporates the input through the input section 26 from a user, and requires acquisition of a license from a license server 4 corresponding to the input.

[0186] Corresponding to this demand, a license server 4 presents a countervalue required in order to buy a license so that it may mention later with reference to the flow chart of drawing 33 (step S242 of drawing 33). Then, in step S223, CPU21 of a client 1 will output and display this on the output section 27, if presentation of the countervalue from a license server 4 is received.

[0187] A user judges whether based on this display, it consents to the shown countervalue, and inputs that decision result from the input section 26 based on that decision result.

[0188] When it judges with CPU21 having judged and consented whether to have consented to the countervalue shown the user in step S224 based on the input from the input section 26, it progresses to step S225 and processing which notifies consent to a license server 4 is performed.

[0189] If this advice of consent is received, a license server 4 will transmit the information showing acquisition of a countervalue, i.e., the mark which described the ownership flag, (step S244 of drawing 33). Then, in step S226, CPU21 of a client 1 will perform processing which embeds the received mark to contents in step S227, if the mark from a license server 4 is received. That is, the mark the ownership flag as shown in drawing 31 was described to be will be recorded as a mark of the contents corresponding to the bought license by this corresponding to contents. Moreover, since it means that, as for CPU21, the message was updated at this time, a digital signature (drawing 25) is also updated and it records on a record medium.

[0190] In step S224, when judged with not consenting to the countervalue shown from

the license server 4, it progresses to step S228 and CPU21 notifies not consenting to the shown countervalue to a license server 4.

[0191] Corresponding to processing of such a client 1, a license server 4 performs processing shown in the flow chart of drawing 33 .

[0192] That is, CPU21 of a license server 4 is first, when the demand of license acquisition is transmitted from a client 1 in step S241 (step S222 of drawing 32), A countervalue required for the acquisition by the target license in reception and step S242 of this is read from the storage section 28, and this is transmitted to a client 1.

[0193] As mentioned above, advice of whether to consent to the countervalue shown from the client 1 to the countervalue shown by doing in this way is transmitted.

[0194] Then, the mark containing the message which progresses to step S244 and expresses the acquisition by the target license is generated, when it judges with having received advice of consent, it judges whether in step S243, CPU21 of a license server 4 received advice of consent from the client 1, it is its own private key and a digital signature is added, and it transmits to a client 1. Thus, the transmitted mark is recorded on corresponding contents in the storage section 28 of a client 1, as mentioned above (step S227 of drawing 32).

[0195] In step S243, when judged with advice of consent not being received from a client 1, processing of step S244 is skipped. That is, in this case, since it means that acquisition processing of a license was not performed eventually, a mark is not transmitted.

[0196] Drawing 34 expresses the example of a configuration of the mark transmitted from a license server 4 to a client 1 in step S244. The mark is constituted in this example by digital signature Sigs (LeafID, Own) generated by that user's leaf ID, the ownership flag (Own), and the list in Leaf ID and the ownership flag based on the private key S of a license server 4.

[0197] In addition, since this mark is effective, when it is copied in the target contents only to a specific user's specific contents, let the mark which accompanies those copied contents be an invalid.

[0198] Thus, contents and a license are separated, and when making a service condition equivalent to a license, it becomes possible to realize service according to the operating condition of each contents.

[0199] Next, grouping is explained. It is called grouping to collect two or more devices and media suitably, and to enable it to deliver and receive contents freely in the one set. Usually, this grouping is performed in the device and media which an individual owns. Although this grouping carried out setting up a group key for every group

conventionally etc. and was performed, it becomes possible [carrying out grouping easily] by matching the same license with two or more devices and media which carry out grouping.

[0200] Moreover, it is also possible to carry out the grouping of each device by registering beforehand. The grouping in this case is explained below.

[0201] In this case, a user needs to register into a server beforehand the certificate of the device made into a grouping object. Registration processing of this certificate is explained with reference to the flow chart of drawing 35 and drawing 36 .

[0202] First, with reference to the flow chart of drawing 35 , registration processing of the certificate of a client (device used as a grouping object) is explained. In step S261, CPU21 of a client 1 draws up its own [as a device made into the object of grouping] certificate. Its own public key is contained in this certificate.

[0203] Next, it progresses to step S262, CPU21 accesses the contents server 3 based on the input from a user's input section 26, and processing which transmits the certificate drawn up by processing of step S261 to the contents server 3 is performed in step S263.

[0204] In addition, as a certificate, what received from the license server 4 can also be used as it is.

[0205] All the devices made into a grouping object perform the above processing.

[0206] Next, with reference to the flow chart of drawing 36 , the registration processing of the certificate of the contents server 3 performed corresponding to registration processing of the certificate of the client 1 of drawing 35 is explained.

[0207] First, in step S271, CPU21 of the contents server 3 will register the certificate into the storage section 28 in step S272, if the certificate transmitted from the client 1 is received.

[0208] The above processing is performed for every device made into a group object. Consequently, as shown in drawing 37 , the certificate of the device which constitutes the group is registered into the storage section 28 of the contents server 3 for every group.

[0209] In the example shown in drawing 37 , a certificate C11 thru/or C14 are registered as a group's 1 certificate. The corresponding public key KP11 thru/or KP14 are contained in these certificate C11 thru/or C14.

[0210] Similarly, as a group's 2 certificate, a certificate C21 thru/or C23 are registered, and the public key KP21 with which these correspond thru/or KP23 are contained.

[0211] If offer of contents is required of the device which belongs to the group from a

user in the condition which constitutes the above groups that the certificate was registered for every device, the contents server 3 will perform processing shown in the flow chart of drawing 38 .

[0212] First, in step S281, CPU21 of the contents server 3 performs processing which verifies the certificate which belongs to the group among the certificates memorized by the storage section 28.

[0213] This verification processing is performed by following EKB using a tag based on the leaf ID contained in the certificate of each device, as explained with reference to drawing 29 and drawing 30 . EKB is distributed also to the contents server 3 from the license server 4. The certificate [RIBOKU / certificate / this verification processing] is excepted.

[0214] In step S282, CPU21 of the contents server 3 chooses the confirmed certificate as a result of verification processing of step S281. And in step S283, CPU21 enciphers a contents key with each public key of the certificate of each device chosen by processing of step S282. In step S284, CPU21 transmits with contents the contents key enciphered by processing of step S283 to each device of the target group.

[0215] Supposing RIBOKU [the certificate C14] among the groups 1 shown in drawing 37 , encryption data as been processing of step S283, for example, shown in drawing 39 will be generated.

[0216] That is, in the example of drawing 39 , the contents key Kc is enciphered with the public key KP11 of a certificate C11, the public key KP12 of a certificate C12, or the public key KP13 of a certificate C13.

[0217] Corresponding to processing as shown in drawing 38 of the contents server 3, the device (client) of each group who receives offer of contents performs processing shown in the flow chart of drawing 40 .

[0218] First, in step S291, CPU21 of a client 1 receives the contents which the contents server 3 has transmitted by processing of step S284 of drawing 38 with a contents key. Contents are enciphered with the contents key Kc, and the contents key is enciphered with the public key which each device holds, as mentioned above (drawing 39).

[0219] Then, in step S292, CPU21 decodes and acquires the contents key addressed to it with its own private key. [who received by processing of step S291] And decode processing of contents is performed using the acquired contents key.

[0220] For example, using its own [corresponding to a public key KP11] private key, the device corresponding to the certificate C11 shown in the example of drawing 39

decodes the code of the contents key Kc, and acquires the contents key Kc. And contents are further decoded using the contents key Kc.

[0221] Same processing is performed also in the device corresponding to certificates C12 and C13. Since the contents key Kc enciphered using its own public key is not sent along with contents, the device of the certificate [RIBOKU / certificate] C14 cannot decode the contents key Kc, therefore cannot decode contents using the contents key Kc.

[0222] Although it was made to perform grouping above to the contents key (namely, contents), it is also possible to perform grouping to a license key (license).

[0223] Grouping becomes possible, without using a special group key and ICV (Integrity Check Value) mentioned later as mentioned above. This grouping is fit for applying to a small-scale group.

[0224] In this invention, it is supposed that it is possible to check out check in it at it, it, or to also copy a license. However, these processings are performed based on the rule defined by SDMI.

[0225] Next, with reference to the flow chart of drawing 41 and drawing 42 , check-out processing of the license by such client is explained.

[0226] First, processing of the client which checks out a license to other clients with reference to the flow chart of drawing 41 is explained. First, in step S301, CPU21 of a client 1 reads the count N1 of check-out of the license for check-out. Since this count of check-out is written in the service condition shown in drawing 8 , it is read in this service condition.

[0227] Next, in step S302, CPU21 reads too the count N2 of the maximum check-out of the license for check-out in the service condition of a license.

[0228] And in step S303, CPU21 measures the count N1 of check-out read by processing of step S301, and the count N2 of the maximum check-out read by processing of step S302, and judges whether the count N1 of check-out is larger than the count N2 of the maximum check-out.

[0229] When it judges that the count N1 of check-out is smaller than the count N2 of the maximum check-out, it progresses to step S304, and CPU21 acquires the leaf key of the equipment (client of a check-out place) of the other party from the equipment of partner each, and stores the leaf key in the check-out list of the storage section 28 corresponding to the license ID now made applicable to check-out.

[0230] Next, in step S305, only 1 increments the value of the count N1 of check-out of the license in which CPU21 was read by processing of step S301. In step S306, CPU21 calculates ICV based on the message of a license. About this ICV, it mentions

later with reference to drawing 46 thru/or drawing 50 . It becomes possible to prevent the alteration of a license using ICV.

[0231] Next, CPU21 enciphers using its own public key, and makes ICV calculated by the license for check-out, and processing of step S306 output and copy to the equipment of the other party with EKB and a certificate in step S307. Furthermore, CPU21 makes ICV calculated by processing of step S306 remember it to be the leaf key of other party equipment in the check list of the storage section 28 in step S308 corresponding to License ID.

[0232] In step S303, when it judges that the count N1 of check-out is not smaller than the count N2 of the maximum check-out (for example, equal), since check-out is performed, only the count already permitted cannot check out any more. Then, it progresses to step S309 and CPU21 performs error processing. That is, check-out processing will be performed in this case.

[0233] Next, with reference to the flow chart of drawing 42 , check-out processing of drawing 41 explains processing of the client which receives check-out of a license.

[0234] First, in step S321, its own leaf key is transmitted to other party equipment (client 1 which checks out a license). This leaf key is memorized by the client of the other party in step S304 corresponding to License ID.

[0235] Next, in step S322, CPU21 receives this, when the license and ICV which were enciphered from the client 1 of the other party have been transmitted with EKB and a certificate. That is, this license, ICV and EKB, and a certificate are transmitted from the equipment of the other party by processing of step S307 of drawing 41 .

[0236] CPU21 makes the storage section 28 memorize the license, ICV and EKB, and the certificate which were received by processing of step S322 in step S323.

[0237] When using the license from which the carrier beam client 1 received check-out for check-out of a license as mentioned above and reproducing predetermined contents, processing shown in the flow chart of drawing 43 is performed.

[0238] That is, in step S341, CPU21 of a client 1 calculates first ICV of the contents as which playback was specified by the user through the input section 26. And CPU21 makes ICV which is memorized by the storage section 28 and which is enciphered decode in step S342 based on the public key contained in the certificate.

[0239] Next, in step S343, CPU21 judges whether ICV now calculated by processing of step S341 and ICV which reading appearance was carried out [ICV] by processing of step S342, and was decoded are in agreement. The license will be altered when both are in agreement. Then, it progresses to step S344 and CPU21 performs

processing which reproduces corresponding contents.

[0240] On the other hand, in step S343, when judged with two ICV(s) not being in agreement, a license has a possibility that it may be altered. For this reason, it progresses to step S345 and CPU21 performs error processing. That is, at this time, contents can be reproduced using that license.

[0241] Next, processing of the client which receives check-in of the license once checked out to other clients as mentioned above is explained with reference to the flow chart of drawing 44 .

[0242] First, in step S361, CPU21 acquires the leaf key of the equipment (client 1 which returns a license (check-in)) of the other party, and ID of the license for check-in. Next, in step S362, he judges whether CPU21 is the license which the license for [which was acquired at step S361] check-in checked out to other party equipment. This judgment is performed based on ICV memorized by processing of step S308 of drawing 41 , a leaf key, and License ID. That is, when it is judged and memorized whether the leaf key acquired at step S361 and Licenses ID and ICV are memorized during the check-out list, it is judged with it being the license which he checked out.

[0243] When he checks [a license] out, in step S363, CPU21 requires deletion of the license of the equipment of the other party, EKB, and a certificate. Based on this demand, the equipment of the other party performs deletion of a license, EKB, and a certificate so that it may mention later (step S383 of drawing 45).

[0244] In step S364, since the once checked-out license has checked in at CPU21 again, only 1 carries out the decrement of the count N1 of check-out of the license.

[0245] In step S365, it judges whether CPU21 has checked out other licenses to the equipment of the other party, and when other licenses which he has still checked out do not exist, it progresses to step S366 and CPU21 deletes the storage in the check-out list as a device for check-in of the equipment of the other party. On the other hand, in step S365, since check-in of other licenses may be received when judged with other licenses which he has checked out to the equipment of the other party existing, processing of step S366 is skipped.

[0246] In step S362, when it judges that the license made applicable to check-in is not a license which he checked out to other party equipment, CPU21 progresses to step S367, and performs error processing. That is, in this case, since it will not be the license he has jurisdiction [license], check-in processing is not performed.

[0247] when a user copies a license unjustly, the value of ICV memorized differs from the value of ICV calculated based on the license acquired by processing of step S361

-- he can come out and check in.

[0248] Drawing 45 expresses processing of the client at which the license which he has is made to check in to the client which performs check-in processing of the license shown in the flow chart of drawing 44 .

[0249] In step S381, CPU21 of a client 1 transmits ID of the license a leaf key and for check-in to the equipment (client 1 which performs processing shown in the flow chart of drawing 44) of the other party. As mentioned above, in step S361, the equipment of the other party acquires this leaf key and License ID, and performs authentication processing of the license for check-in in step S362 based on it.

[0250] In step S382, CPU21 of a client 1 judges whether deletion of a license was required from the equipment of the other party. Namely, when a license is a license for [just] check-in, as mentioned above, as for the equipment of the other party, deletion of a license, EKB, and a certificate is required by processing of step S363. Then, when this demand is received, it progresses to step S383 and CPU21 deletes a license, EKB, and a certificate. That is, since DEKURIMENDO [this client 1 will be in the condition that that license cannot be used henceforth and / processing of step S364 of drawing 44 / the count N1 of check-out / 1], it means that check-in was completed by this.

[0251] In step S382, when judged with deletion of a license not being demanded from the equipment of the other party, it progresses to step S384 and error processing is performed. That is, check-in will not be possible for the reasons of the values of ICV differing in this case.

[0252] Although check-in and check-out were explained above, it is possible similarly to make it also make a license MUBU [copy or].

[0253] Next, in order to prevent the alteration of a license (the same is said of contents), the integrity check value (ICV) of a license is generated, and it matches with a license, and count of ICV explains the processing configuration which judges the existence of a license alteration.

[0254] The integrity check value (ICV) of a license is calculated using the Hash Function to a license, and is calculated by ICV=hash (Kicv, L1 and L2, ...). Kicv is an ICV generation key. L1 and L2 are the information on a license, and the message authenticator (MAC:Message authentication Code) of the critical information of a license is used.

[0255] The example of MAC value generation using a DES cipher-processing configuration is shown in drawing 46 . the (target message as shown in the configuration of drawing 46 -- a 8-byte unit -- dividing -- the divided message is

hereafter set to) M1, M2, ..., MN -- the exclusive OR of M1 is first carried out to initial value (IV) by operation part 24-1A (the result is set to I1). Next, I1 is put into DES encryption section 24-1B, and it enciphers using a key (hereafter referred to as K1) (an output is set to E1). Continuously, the exclusive OR of E1 and M2 is carried out by operation part 24-2A, the output I2 is put in to DES encryption section 24-2B, and it enciphers using a key K1 (output E2). Hereafter, this is repeated and encryption processing is performed to all messages. EN which appeared from DES encryption section 24-NB in the last serves as a message authenticator (MAC (Message Authentication Code)).

[0256] The integrity check value (ICV) of a license is generated by the MAC value and ICV generation key of such a license with the application of a Hash Function. For example, if it will be guaranteed that there is no alteration in a license if ICV generated to the license generate time is compared with ICV newly generated based on the license and the same ICV is obtained, and ICV(s) differ, it will be judged with there having been an alteration.

[0257] Next, the configuration which sends Kicv which is the integrity check value (ICV) generation key of a license by the above-mentioned validation key block is explained. That is, it is the example used as the integrity check value (ICV) generation key of a license of the encryption message data based on EKB.

[0258] When a license common to two or more devices is sent to drawing 47 and drawing 48 , the example of a configuration which distributes the integrity check value generation key Kicv for verifying the existence of an alteration of those licenses by the validation key block (EKB) is shown. Drawing 47 shows the example which distributes the check value generation key Kicv which can be decoded to devices 0, 1, 2, and 3, and drawing 48 shows the example which carries out RIBOKU (abatement) of devices 0, 1, and 2 and the device 3 in three, and distributes the check value generation key Kicv which can be decoded only to devices 0, 1, and 2.

[0259] drawing 47 -- an example -- **** -- updating -- a node -- a key -- K -- (-- t --) -- 00 -- a check -- a value -- generation -- a key -- Kicv -- having enciphered -- data -- Enc (K (t) 00 Kicv) -- a device -- zero -- one -- two -- three -- setting -- each -- having -- a node -- a key -- a leaf -- a key -- using -- updating -- having had -- a node -- a key -- K -- (-- t --) -- 00 -- decode -- being possible -- validation -- a key block (EKB) -- generating -- distributing . each -- a device -- drawing 47 -- right-hand side -- being shown -- as -- first -- EKB -- processing (decode) -- carrying out -- things -- updating -- having had -- a node -- a key -- K -- (-- t --) -- 00 -- acquiring -- next -- having acquired -- a node -- a key -- K --

(-- t --) -- 00 -- using -- enciphering -- having had -- a check -- a value -- generation -- a key -- Enc (K (t) 00 Kicv) -- decoding -- a check -- a value -- generation -- a key -- Kicv -- obtaining -- things -- being possible -- ** -- becoming .

[0260] others -- a device -- four -- five -- six -- seven ... being the same -- validation -- a key block (EKB) -- receiving -- even if -- self -- holding -- a node -- a key -- a leaf -- a key -- **** -- EKB -- processing -- updating -- having had -- a node -- a key -- K -- (-- t --) -- 00 -- being unacquirable -- since -- a safely just device -- only receiving -- a check value generation key -- it can send .

[0261] On the other hand, the example of drawing 48 is other groups' member, i.e., the example which generated and distributed the validation key block (EKB) which can be decoded only to devices 0, 1, and 2, noting that RIBOKU (abatement) of the device 3 is carried out by leakage of a key in the group enclosed with the dotted-line frame of drawing 12 . The data Enc (K (t) 00 Kicv) which enciphered the check value generation key (Kicv) as the validation key block (EKB) shown in drawing 48 by the node key (K (t) 00) are distributed.

[0262] The decode procedure is shown in the right-hand side of drawing 48 . Devices 0, 1, and 2 acquire an updating node key (K (t) 00) from the received validation key block first by decode processing using the leaf key or node key which self holds. Next, the check value generation key Kicv is acquired by the decode by K(t)00.

[0263] other groups' devices 4, 5, and 6 shown in drawing 12 -- even if ... receives this same data (EKB), an updating node key (K (t) 00) is unacquirable using the leaf key and node key which self holds. Also in the device [RIBOKU / device / similarly] 3, by the leaf key and node key which self holds, an updating node key (K (t) 00) cannot be acquired, but only the device which has just access becomes possible [decoding and using a check value generation key].

[0264] Thus, if delivery of the check value generation key using EKB is used, it will become possible to distribute the check value generation key whose decode lessened the amount of data and only the just rightful claimant enabled at insurance.

[0265] The illegal copy of EKB and an encryption license can be eliminated by using the integrity check value (ICV) of such a license. For example, as shown in drawing 49 A, there are media 1 which stored the license L1 and the license L2 with the validation key block (EKB) which can acquire each license key, and the case where this is copied to media 2 as it was is assumed. The copy of EKB and an encryption license will be possible and can use this with the device which can decode EKB.

[0266] In the example shown in drawing 49 B, it considers as the configuration which

matches with the license justly stored in each media, and stores an integrity check value (ICV (L1, L2)). In addition, (ICV (L1, L2)) shows ICV=hash (Kicv, L1, L2) which is the integrity check value of the license calculated by using a Hash Function for license L1 and license L2. In the configuration of drawing 49 B, license 1 and license 2 are justly stored in media 1, and the integrity check value (ICV (L1, L2)) generated based on the license L1 and the license L2 is stored in them. Moreover, license 1 is justly stored in media 2, and the integrity check value (ICV (L1)) generated based on the license L1 is stored in them.

[0267] In this configuration, supposing it copies {EKB and license 2} which were stored in media 1 to media 2, unlike Kicv (L1) which ICV (L1, L2) will be generated and is stored in media 2 by media 2 if a license check value is newly generated, it will become clear that storing of the new license by an alteration or the unjust copy of a license was performed. In the device which reproduces media, an ICV check is performed to the step before a playback step, and coincidence of Generation ICV and Storing ICV is distinguished, and when not in agreement, it becomes possible to prevent playback of a license of an illegal copy by considering as the configuration which does not perform playback.

[0268] Furthermore, in order to raise safety, it is good also as a configuration generated based on the data which rewrote the integrity check value (ICV) of a license and include a counter. That is, it considers as the configuration calculated by ICV=hash (Kicv, counter+1, L1 and L2, ...). Here, a counter (counter+1) is set up as a value by which one increment is carried out for every rewriting of ICV. In addition, a counter value needs to consider as the configuration stored in secure memory.

[0269] Furthermore, in the configuration which cannot store the integrity check value (ICV) of a license in the same media as a license, it is good also as a configuration which stores the integrity check value (ICV) of a license on media with an another license.

[0270] For example, when it stores a license in the media by which copy preventive measures, such as the ReadOnly media and the usual MO, are not taken, if an integrity check value (ICV) is stored in the same media, rewriting of ICV may be made by the inaccurate user, and there is a possibility that the safety of ICV cannot be maintained. In such a case, safe management of ICV and the alteration check of a license are attained by storing ICV in the safe media on a host machine, and considering as the configuration which uses ICV for copy control (for example, check-in/check-out, move) of a license.

[0271] This example of a configuration is shown in drawing 50 . In drawing 50 , it is the

example which the license 1 thru/or the license 3 were stored in the media 2201 by which copy preventive measures, such as the ReadOnly media and the usual MO, are not taken, stored the integrity check value (ICV) about these licenses in the safe media 2202 on the host machine with which it is not permitted that a user accesses freely, and prevented rewriting of the unjust integrity check value (ICV) by the user. As such a configuration, in case the device equipped with media 2201 performs playback of media 2201, it can prevent playback of PC which is a host machine, the configuration which performs the check of ICV in a server and judges reproductive propriety then an unjust copy license, or an alteration license.

[0272] The client to which this invention is applied can be used as PDA (Personal Digital Assistants), a portable telephone, a game terminal, etc. in addition to the so-called personal computer.

[0273] When performing a series of processings with software, the program which constitutes the software is installed in a general-purpose personal computer etc. from a network or a record medium possible [performing various kinds of functions] by installing the computer built into the hardware of dedication, or various kinds of programs.

[0274] As this record medium is shown in drawing 2 , apart from the body of equipment, are distributed in order to provide a user with a program. The magnetic disk 41 (a floppy disk is included) with which the program is recorded, an optical disk 42 (CD-ROM (Compact Disk-ReadOnly Memory) --) DVD (Digital Versatile Disk) is included. It is not only constituted by the package media which consist of a magneto-optic disk 43 (MD (Mini-Disk) is included) or semiconductor memory 44, but It consists of ROM22 with which a user is provided in the condition of having been beforehand included in the body of equipment and on which the program is recorded, a hard disk contained in the storage section 28.

[0275] In addition, in this description, even if the processing serially performed in accordance with the sequence that the step which describes the program recorded on a record medium was indicated is not of course necessarily processed serially, it is a juxtaposition thing also including the processing performed according to an individual.

[0276] Moreover, in order for the program which performs processing relevant to security to prevent analyzing the processing, it is desirable to encipher the program itself. For example, the processing which performs cipher processing etc. can constitute the program as a tamper REJISUTANTO module.

[0277] Moreover, since the license which carries out utilization authorization of the contents is specified, the information indicated by the header of contents may not be

the license ID which identifies a license uniquely. It is the information which specifies the license to utilization of contents to be licensed [ID] in the above-mentioned example, and a certain license is the information which specifies the contents which permit utilization, and it is the information which identifies the license demanded by license demand from a client 1. The list of the various attribute information about the contents of contents is indicated by contents, and you may make it indicate with a license the conditional expression of the contents in which utilization authorization is carried out by the license. In this case, the attribute information included in contents is the information which specifies the license to which utilization of those contents is permitted, and the conditional expression contained in a license is the information as which that license specifies the contents which permit utilization, and serves as information from which License ID discriminates a license uniquely. When it does in this way, it becomes possible to match two or more licenses with one contents, and a license can be published flexibly.

[0278] Moreover, in this description, a system expresses the whole equipment constituted by two or more equipments.

[0279]

[Effect of the Invention] It enables it to distribute freely the data which were enciphered according to the program to the information processor of this invention and an approach, a license server, and a list like the above, and by having enabled it to use contents by acquiring a license separately, without barring the negotiation of contents, copyright can be protected and suitable dues can be collected.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the contents offer structure of a system which applied this invention.

[Drawing 2] It is the block diagram showing the configuration of the client of drawing 1 .

[Drawing 3] It is a flow chart explaining download processing of the contents of the client of drawing 1 .

[Drawing 4] It is a flow chart explaining contents offer processing of the contents server of drawing 1 .

[Drawing 5] It is drawing showing the example of the format in step S26 of drawing 4 .

[Drawing 6] It is a flow chart explaining contents regeneration of the client of drawing 1 .

[Drawing 7] It is a flow chart explaining the detail of license acquisition processing of step S43 of drawing 6 .

[Drawing 8] It is drawing showing the configuration of a license.

[Drawing 9] It is a flow chart explaining processing of license offer of the license server of drawing 1 .

[Drawing 10] It is a flow chart explaining the detail of the license update process in step S45 of drawing 6 .

[Drawing 11] It is a flow chart explaining a license update process of the license server of drawing 1 .

[Drawing 12] It is drawing explaining the configuration of a key.

[Drawing 13] It is drawing explaining a category node.

[Drawing 14] It is drawing showing the example of a response of a node and a device.

[Drawing 15] It is drawing explaining the configuration of a validation key block.

[Drawing 16] It is drawing explaining utilization of a validation key block.

[Drawing 17] It is drawing showing the example of a format of a validation key block.

[Drawing 18] It is drawing explaining the configuration of the tag of a validation key block.

[Drawing 19] It is drawing explaining decode processing of the contents using DNK.

[Drawing 20] It is drawing showing the example of a validation key block.

[Drawing 21] It is drawing explaining the assignment to one device of two or more contents.

[Drawing 22] It is drawing explaining the category of a license.

[Drawing 23] It is a flow chart explaining ripping processing of a client.

[Drawing 24] It is drawing explaining the configuration of a watermark.
[Drawing 25] It is drawing showing the example of a format of contents.
[Drawing 26] It is drawing showing the example of a public key certificate.
[Drawing 27] It is drawing explaining distribution of contents.
[Drawing 28] It is a flow chart explaining check-out processing of the contents of a client.
[Drawing 29] It is drawing explaining the example which follows the validation key block by the tag.
[Drawing 30] It is drawing showing the example of a configuration of a validation key block.
[Drawing 31] It is drawing explaining the configuration of a mark.
[Drawing 32] It is a flow chart explaining license acquisition processing of a client.
[Drawing 33] It is a flow chart explaining license acquisition processing of a license server.
[Drawing 34] It is drawing showing the example of a configuration of a mark.
[Drawing 35] It is a flow chart explaining registration processing of the certificate of a client.
[Drawing 36] It is a flow chart explaining certificate registration processing of a contents server.
[Drawing 37] It is drawing showing the example of a group's certificate.
[Drawing 38] It is a flow chart explaining processing of a contents server in case grouping is performed.
[Drawing 39] It is drawing showing the example of encryption of a contents key.
[Drawing 40] It is a flow chart explaining processing of the client belonging to a group.
[Drawing 41] It is the flow chart which explains to other clients processing of the client which checks out a license.
[Drawing 42] It is a flow chart explaining processing of the client which receives check-out of a license from other clients.
[Drawing 43] It is the flow chart which explains regeneration of a carrier beam client for check-out of a license.
[Drawing 44] It is a flow chart explaining processing of the client which receives check-in of a license from other clients.
[Drawing 45] It is the flow chart which explains to other clients processing of the client which checks in at a license.
[Drawing 46] It is drawing explaining generation of MAC.
[Drawing 47] It is a flow chart explaining decode processing of an ICV generation key.

[Drawing 48] It is drawing explaining other decode processings of an ICV generation key.

[Drawing 49] It is drawing explaining management of the copy of the license by ICV.

[Drawing 50] It is drawing explaining management of a license.

[Description of Notations]

The 1-1 and 1-2 input section, 27 output section, the 28 storage section, the 29 communications department Client 2 Internet 3 Contents server 4 License server 5 Accounting server 20 Timer 21CPU 24 Encryption decode section 25 Codec section 26

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－359616
（P2002－359616A）
(43)公開日　平成14年12月13日(2002.12.13)

(51)Int.Cl.[7]		識別記号	FI			テーマコード(参考)
Ｈ０４Ｌ	9/08		Ｇ０９Ｃ	5/00		５Ｊ１０４
Ｇ０９Ｃ	5/00		Ｈ０４Ｌ	9/00	ＺＥＣ	
Ｈ０４Ｌ	9/00	ＺＥＣ			６０１Ｂ	
	9/32				６７５Ｂ	

審査請求　未請求　請求項の数14　ＯＬ　（全 35 頁）

(21)出願番号　特願2002－28915(P2002－28915)

(22)出願日　平成14年２月６日(2002.2.6)

(31)優先権主張番号　特願2001－33114(P2001－33114)
(32)優先日　平成13年２月９日(2001.2.9)
(33)優先権主張国　日本（ＪＰ）
(31)優先権主張番号　特願2001－94803(P2001－94803)
(32)優先日　平成13年３月29日(2001.3.29)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　田中　浩一
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　河上　達
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　100082131
弁理士　稲本　義雄

最終頁に続く

(54)【発明の名称】　情報処理装置および方法、ライセンスサーバ、並びにプログラム

(57)【要約】
【課題】　コンテンツの配布を自由に行うことができ、許可されたユーザのみがコンテンツを利用できるようにする。
【解決手段】　クライアントは暗号化されたコンテンツをコンテンツサーバから受け取る。コンテンツのヘッダにはそのコンテンツを利用するとき必要とされるライセンスを特定するためのライセンス特定情報が記述されており、クライアントはライセンス特定情報を元にライセンスサーバにライセンスを要求する。ライセンスサーバは、ライセンス要求を受け取ると、課金処理を行った後、該当するライセンスをクライアントに送信する。クライアントはライセンスを保持していることを条件として、コンテンツを復号し再生する。

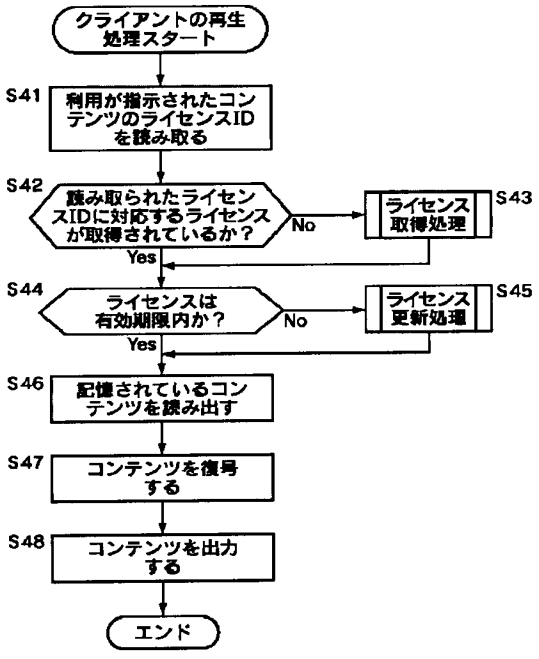


図6

1

【特許請求の範囲】

【請求項１】　ライセンスを保持していることを条件としてコンテンツの利用を許可する情報処理装置において、

当該コンテンツを利用許可する前記ライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報とを含む前記コンテンツを記憶するコンテンツ記憶手段と、

利用許可される前記コンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するライセンス記憶手段と、前記コンテンツを利用許可することができるライセンスが前記ライセンス記憶手段に記憶されているか否かを判定する判定手段と、

前記判定手段によりライセンスが記憶されていると判断されたことを条件として前記コンテンツのコンテンツデータを復号する復号手段とを備えることを特徴とする情報処理装置。

【請求項２】　前記情報処理装置は更に、

ライセンスサーバにライセンスを識別するためのライセンス識別情報を含むライセンス要求を送信する送信手段と、

ライセンスサーバによって送信されたライセンスを受信する受信手段とを備え、

前記受信手段により受信されたライセンスは前記ライセンス記憶手段に記憶されることを特徴とする請求項１記載の情報処理装置。

【請求項３】　前記コンテンツデータはテキストデータ、画像データ、音声データ、動画データあるいはそれらを組み合わせたデータであり、

前記復号手段により復号されたコンテンツデータを再生する再生手段を更に備えることを特徴とする請求項１記載の情報処理装置。

【請求項４】　前記鍵情報はEKB（Enabling Key Block）を含み、

前記情報処理装置は更にデバイスノードキーを記憶するデバイスノードキー記憶手段を備え、

前記復号手段は前記デバイスノードキー記憶手段に記憶されている前記デバイスノードキーを用いて前記EKB（Enabling Key Block）を復号処理し得られたルートキーを用いて前記暗号化されたコンテンツデータを復号することを特徴とする請求項１記載の情報処理装置。

【請求項５】　前記鍵情報は更に前記EKB（Enabling Key Block）のルートキーによって暗号化されたコンテンツキーを含み、

前記コンテンツデータは前記コンテンツキーにより暗号化されており、

前記復号手段は前記デバイスノードキー記憶手段に記憶されている前記デバイスノードキーを用いて前記EKB（Enabling Key Block）を復号処理し得られたルートキー

2

を用いて復号された前記コンテンツキーを用いて前記暗号化されたコンテンツデータを復号することを特徴とする請求項４記載の情報処理装置。

【請求項６】　前記ライセンスは更に、当該ライセンスによって利用可能となるコンテンツの使用条件を示す使用条件情報を含むことを特徴とする請求項１記載の情報処理装置。

【請求項７】　前記ライセンスは更に、ライセンスサーバの秘密鍵によりなされた電子署名を含むことを特徴とする請求項１記載の情報処理装置。

【請求項８】　前記情報処理装置は、更に情報処理装置を識別する端末識別情報を記憶する端末識別情報記憶手段を備え、前記ライセンス要求は更に、端末識別情報記憶手段に記憶されている前記端末識別情報を含み、

前記受信手段により受信された前記ライセンスは更に、前記端末識別情報を含み、

前記判定手段は、前記ライセンスに含まれる前記端末識別情報と前記端末識別情報記憶手段に記憶されている前記端末識別情報とを比較し、両者が一致している場合に限り、当該ライセンスを前記コンテンツの利用を許可できるライセンスであると判定することを特徴とする請求項２記載の情報処理装置。

【請求項９】　ライセンスを保持していることを条件としてコンテンツの利用を許可する情報処理方法であって、

当該コンテンツを利用許可する前記ライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報と、を含むコンテンツを記憶するステップと、

当該ライセンスによって利用許可される前記コンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するステップと、

前記コンテンツを利用許可することができるライセンスが前記ライセンス記憶手段に記憶されているか否かを判定するステップと、

前記判定手段によりライセンスが記憶されていると判断されたことを条件として前記コンテンツのコンテンツデータを復号するステップとを含むことを特徴とする情報処理方法。

【請求項１０】　ライセンスを保持していることを条件としてコンテンツの利用を許可する処理をコンピュータに実行させるプログラムであって、

当該コンテンツを利用許可する前記ライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報と、を含むコンテンツを記憶するステップと、

当該ライセンスによって利用許可される前記コンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するステップと、

前記コンテンツを利用許可することができるライセンス

が前記ライセンス記憶手段に記憶されているか否かを判定するステップと、
前記判定手段によりライセンスが記憶されていると判断されたことを条件として前記コンテンツのコンテンツデータを復号するステップとをコンピュータに実行させるプログラム。
【請求項１１】　前記プログラムあるいはその一部が暗号化されていることを特徴とする請求項１０記載のプログラム。
【請求項１２】　コンテンツの利用を許可するライセンスを発行するライセンスサーバにおいて、
当該ライセンスによって利用許可される前記コンテンツを特定するためのコンテンツ特定情報と、情報処理装置を識別する端末識別情報を含む前記ライセンスを記憶するライセンス記憶手段と情報処理装置から送信された、ライセンスを識別するライセンス識別情報を含むライセンス要求を受信する受信手段と、
前記ライセンス要求に含まれる前記ライセンス識別情報に対応する前記ライセンスを前記ライセンス記憶手段から抽出する抽出手段と、
前記抽出手段により抽出された前記ライセンスに前記端末識別情報を付加する処理手段と、
ライセンスサーバの秘密鍵を用いて、前記処理手段により端末識別情報を付加されたライセンスに電子署名を付加する署名手段と、
前記署名手段により署名されたライセンスを前記ライセンス要求を送信した情報処理装置に送信する送信手段とを備えることを特徴とするライセンスサーバ。
【請求項１３】　コンテンツの利用を許可するライセンスを発行する情報処理方法であって、
当該ライセンスによって利用許可される前記コンテンツを特定するためのコンテンツ特定情報と、情報処理装置を識別する端末識別情報を含む前記ライセンスを記憶するステップと、
情報処理装置から送信された、ライセンスを識別するライセンス識別情報を含むライセンス要求を受信するステップと、
前記ライセンス要求に含まれる前記ライセンス識別情報に対応する前記ライセンスを前記ライセンス記憶手段から抽出するステップと、
前記抽出手段により抽出された前記ライセンスに前記端末識別情報を付加するステップと、
ライセンスサーバの秘密鍵を用いて、前記処理手段により端末識別情報を付加されたライセンスに電子署名を付加するステップと、
前記署名手段により署名されたライセンスを前記ライセンス要求を送信した情報処理装置に送信するステップとを含むことを特徴とする情報処理方法。
【請求項１４】　コンテンツの利用を許可するライセンスを発行する処理処理をコンピュータに実行させるプログラムであって、
当該ライセンスによって利用許可される前記コンテンツを特定するためのコンテンツ特定情報と、情報処理装置を識別する端末識別情報を含む前記ライセンスを記憶するステップと、
情報処理装置から送信された、ライセンスを識別するライセンス識別情報を含むライセンス要求を受信するステップと、
前記ライセンス要求に含まれる前記ライセンス識別情報に対応する前記ライセンスを前記ライセンス記憶手段から抽出するステップと、
前記抽出手段により抽出された前記ライセンスに前記端末識別情報を付加するステップと、
ライセンスサーバの秘密鍵を用いて、前記処理手段により端末識別情報を付加されたライセンスに電子署名を付加するステップと、
前記署名手段により署名されたライセンスを前記ライセンス要求を送信した情報処理装置に送信するステップとをコンピュータに実行させるプログラム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、情報処理装置および方法、ライセンスサーバ、並びにプログラムに関し、特に、著作権者からライセンスを受けていないコンテンツが不正にコピーされ、利用されるのを防止することができるようにした、情報処理装置および方法、ライセンスサーバ、並びにプログラムに関する。
【０００２】
【従来の技術】最近、インターネットを介して、ユーザが、自分自身が保持している音楽データを他のユーザに提供し、自分自身が保持していない音楽データを他のユーザから提供を受けるようにして、複数のユーザが無料で音楽データを交換しあうシステムが実現されている。
【０００３】このようなシステムでは、理論的には、１つの音楽、その他のコンテンツが存在すれば、他の全てのユーザが、それを利用することが可能となり、多くのユーザがコンテンツを購入しなくなるため、コンテンツに関する著作権者は、著作物としてのコンテンツが売れないため、著作物の販売に伴い、本来受け取ることが可能な著作物の利用に関するロイヤリティを受け取る機会を失うことになる。
【０００４】
【発明が解決しようとする課題】そこで、コンテンツの流通を妨げることなく、不正に利用されることを防止することが、社会的に要請されている。
【０００５】本発明はこのような状況に鑑みてなされたものであり、コンテンツが不正に利用されるのを確実に防止することができるようにするものである。
【０００６】
【課題を解決するための手段】本発明の情報処理装置

は、コンテンツを利用許可するために必要なライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報とを含むコンテンツを記憶するコンテンツ記憶手段と、利用許可されるコンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するライセンス記憶手段と、コンテンツを利用許可することができるライセンスがライセンス記憶手段に記憶されているか否かを判定する判定手段と、判定手段によりライセンスが記憶されていると判断されたことを条件としてコンテンツのコンテンツデータを復号する復号手段とを備えることを特徴とする。

【０００７】情報処理装置は更に、ライセンスサーバにライセンスを識別するためのライセンス識別情報を含むライセンス要求を送信する送信手段と、ライセンスサーバによって送信されたライセンスを受信する受信手段とを備え、受信手段により受信されたライセンスはライセンス記憶手段に記憶されるようにすることができる。

【０００８】コンテンツデータはテキストデータ、画像データ、音声データ、動画データあるいはそれらを組み合わせたデータであり、復号手段により復号されたコンテンツデータを再生する再生手段を更に備えるようにすることができる。

【０００９】鍵情報はEKB（Enabling Key Block）を含み、情報処理装置は更にデバイスノードキーを記憶するデバイスノードキー記憶手段を備え、復号手段はデバイスノードキー記憶手段に記憶されているデバイスノードキーを用いてEKB（EnablingKey Block）を復号処理し得られたルートキーを用いて暗号化されたコンテンツデータを復号するようにすることができる。

【００１０】鍵情報は更にEKB（Enabling Key Block）のルートキーによって暗号化されたコンテンツキーを含み、コンテンツデータはコンテンツキーにより暗号化されており、復号手段はデバイスノードキー記憶手段に記憶されているデバイスノードキーを用いてEKB（Enabling Key Block）を復号処理し得られたルートキーを用いて復号されたコンテンツキーを用いて暗号化されたコンテンツデータを復号するようにすることができる。

【００１１】ライセンスは更に、そのライセンスによって利用可能となるコンテンツの使用条件を示す使用条件情報を含むようにすることができる。

【００１２】ライセンスは更に、ライセンスサーバの秘密鍵によりなされた電子署名を含むようにすることができる。

【００１３】情報処理装置は、更に情報処理装置を識別する端末識別情報を記憶する端末識別情報記憶手段を備え、ライセンス要求は更に、端末識別情報記憶手段に記憶されている端末識別情報を含み、受信手段により受信されたライセンスは更に、端末識別情報を含み、判定手段は、ライセンスに含まれる端末識別情報と端末識別情報記憶手段に記憶されている端末識別情報とを比較し、両者が一致している場合に限り、そのライセンスをコンテンツの利用を許可できるライセンスであると判定するようにすることができる。

【００１４】本発明の情報処理方法は、コンテンツを利用許可するライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報と、を含むコンテンツを記憶するステップと、利用許可されるコンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するステップと、コンテンツを利用許可することができるライセンスがライセンス記憶手段に記憶されているか否かを判定するステップと、判定手段によりライセンスが記憶されていると判断されたことを条件としてコンテンツのコンテンツデータを復号するステップとを含むことを特徴とする。

【００１５】本発明のプログラムは、コンテンツを利用許可するライセンスを特定するためのライセンス特定情報と、暗号化されたコンテンツデータと、コンテンツデータを復号するために必要な鍵情報と、を含むコンテンツを記憶するステップと、利用許可されるコンテンツを特定するためのコンテンツ特定情報を含むライセンスを記憶するステップと、コンテンツを利用許可することができるライセンスがライセンス記憶手段に記憶されているか否かを判定するステップと、判定手段によりライセンスが記憶されていると判断されたことを条件としてコンテンツのコンテンツデータを復号するステップとをコンピュータに実行させる。

【００１６】プログラムあるいはその一部が暗号化されているようにすることができる。

【００１７】本発明のライセンスサーバは、許可されるコンテンツを特定するためのコンテンツ特定情報と、情報処理装置を識別する端末識別情報を含むライセンスを記憶するライセンス記憶手段と、情報処理装置から送信された、ライセンスを識別するライセンス識別情報を含むライセンス要求を受信する受信手段と、ライセンス要求に含まれるライセンス識別情報に対応するライセンスをライセンス記憶手段から抽出する抽出手段と、抽出手段により抽出されたライセンスに端末識別情報を付加する処理手段と、ライセンスサーバの秘密鍵を用いて、処理手段により端末識別情報を付加されたライセンスに電子署名を付加する署名手段と、署名手段により署名されたライセンスをライセンス要求を送信した情報処理装置に送信する送信手段とを備えることを特徴とする。

【００１８】本発明の情報処理方法は、利用許可されるコンテンツを特定するためのコンテンツ特定情報と、情報処理装置を識別する端末識別情報を含むライセンスを記憶するステップと、情報処理装置から送信された、ライセンスを識別するライセンス識別情報を含むライセンス要求を受信するステップと、ライセンス要求に含まれ

るライセンス識別情報に対応するライセンスをライセンス記憶手段から抽出するステップと、抽出手段により抽出されたライセンスに端末識別情報を付加するステップと、ライセンスサーバの秘密鍵を用いて、処理手段により端末識別情報を付加されたライセンスに電子署名を付加するステップと、署名手段により署名されたライセンスをライセンス要求を送信した情報処理装置に送信するステップとを含むことを特徴とする。

【００１９】本発明の情報処理装置、情報処理方法、並びにプログラムでは、ライセンスを保持していることを条件としてコンテンツを復号し、利用可能にする。

【００２０】本発明のライセンスサーバ、並びに情報処理方法では、特定の情報処理装置でのみ有効なライセンスを発行する。

【００２１】

【発明の実施の形態】図１は、本発明を適用したコンテンツ提供システムの構成を示している。インターネット２には、クライアント１－１，１－２（以下、これらのクライアントを個々に区別する必要がない場合、単にクライアント１と称する）が接続されている。この例においては、クライアントが２台のみ示されているが、インターネット２には、任意の台数のクライアントが接続される。

【００２２】また、インターネット２には、クライアント１に対してコンテンツを提供するコンテンツサーバ３、コンテンツサーバ３が提供するコンテンツを利用するのに必要なライセンスをクライアント１に対して付与するライセンスサーバ４、およびクライアント１がライセンスを受け取った場合に、そのクライアント１に対して課金処理を行う課金サーバ５が接続されている。

【００２３】これらのコンテンツサーバ３、ライセンスサーバ４、および課金サーバ５も、任意の台数、インターネット２に接続される。

【００２４】図２はクライアント１の構成を表している。

【００２５】図２において、CPU（Central Processing Unit）２１は、ROM（Read Only Memory）２２に記憶されているプログラム、または記憶部２８からRAM（Random Access Memory）２３にロードされたプログラムに従って各種の処理を実行する。タイマ２０は、計時動作を行い、時刻情報をCPU２１に供給する。RAM２３にはまた、CPU２１が各種の処理を実行する上において必要なデータなども適宜記憶される。

【００２６】暗号化復号部２４は、コンテンツデータを暗号化するとともに、既に暗号化されているコンテンツデータを復号する処理を行う。コーデック部２５は、例えば、ATRAC（Adaptive Transform Acoustic Coding）３方式などでコンテンツデータをエンコードし、入出力インタフェース３２を介してドライブ３０に接続されている半導体メモリ４４に供給し、記録させる。あるいは、また、コーデック部２５は、ドライブ３０を介して半導体メモリ４４より読み出した、エンコードされているデータをデコードする。

【００２７】半導体メモリ４４は、例えば、メモリスティック（商標）などにより構成される。

【００２８】CPU２１、ROM２２、RAM２３、暗号化復号部２４、およびコーデック部２５は、バス３１を介して相互に接続されている。このバス３１にはまた、入出力インタフェース３２も接続されている。

【００２９】入出力インタフェース３２には、キーボード、マウスなどよりなる入力部２６、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部２７、ハードディスクなどより構成される記憶部２８、モデム、ターミナルアダプタなどより構成される通信部２９が接続されている。通信部２９は、インターネット２を介しての通信処理を行う。通信部２９はまた、他のクライアントとの間で、アナログ信号またはデジタル信号の通信処理を行う。

【００３０】入出力インタフェース３２にはまた、必要に応じてドライブ３０が接続され、磁気ディスク４１、光ディスク４２、光磁気ディスク４３、或いは半導体メモリ４４などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部２８にインストールされる。

【００３１】なお、図示は省略するが、コンテンツサーバ３、ライセンスサーバ４、課金サーバ５も、図２に示したクライアント１と基本的に同様の構成を有するコンピュータにより構成される。そこで、以下の説明においては、図２の構成は、コンテンツサーバ３、ライセンスサーバ４、課金サーバ５などの構成としても引用される。

【００３２】次に、図３のフローチャートを参照して、クライアント１がコンテンツサーバ３からコンテンツの提供を受ける処理について説明する。

【００３３】ユーザが、入力部２６を操作することでコンテンツサーバ３に対するアクセスを指令すると、CPU２１は、ステップＳ１において、通信部２９を制御し、インターネット２を介してコンテンツサーバ３にアクセスさせる。ステップＳ２において、ユーザが、入力部２６を操作して、提供を受けるコンテンツを指定すると、CPU２１は、この指定情報を受け取り、通信部２９から、インターネット２を介してコンテンツサーバ３に、指定されたコンテンツを通知する。図４のフローチャートを参照して後述するように、この通知を受けたコンテンツサーバ３は、暗号化されたコンテンツデータを送信してくるので、ステップＳ３において、CPU２１は、通信部２９を介して、このコンテンツデータを受信すると、ステップＳ４において、その暗号化されているコンテンツデータを記憶部２８を構成するハードディスクに供給し、記憶させる。

【００３４】次に、図４のフローチャートを参照して、クライアント１の以上の処理に対応するコンテンツサーバ３のコンテンツ提供処理について説明する。なお、以下の説明において、図２のクライアント１の構成は、コンテンツサーバ３の構成としても引用される。

【００３５】ステップＳ２１において、コンテンツサーバ３のCPU２１は、インターネット２から通信部２９を介してクライアント１よりアクセスを受けるまで待機し、アクセスを受けたと判定したとき、ステップＳ２２に進み、クライアント１から送信されてきたコンテンツを指定する情報を取り込む。このコンテンツを指定する情報は、クライアント１が、図３のステップＳ２において通知してきた情報である。

【００３６】ステップＳ２３において、コンテンツサーバ３のCPU２１は、記憶部２８に記憶されているコンテンツデータの中から、ステップＳ２２の処理で取り込まれた情報で指定されたコンテンツを読み出す。CPU２１は、ステップＳ２４において、記憶部２８から読み出されたコンテンツデータを、暗号化復号部２４に供給し、コンテンツキーKcを用いて暗号化させる。

【００３７】記憶部２８に記憶されているコンテンツデータは、コーデック部２５により、既にATRAC３方式によりエンコードされているので、このエンコードされているコンテンツデータが暗号化されることになる。

【００３８】なお、もちろん、記憶部２８に予め暗号化した状態でコンテンツデータを記憶させることができる。この場合には、ステップＳ２４の処理は省略することが可能である。

【００３９】次に、ステップＳ２５において、コンテンツサーバ３のCPU２１は、暗号化したコンテンツデータを伝送するフォーマットを構成するヘッダに、暗号化されているコンテンツを復号するのに必要なキー情報（図５を参照して後述するEKBとK_{EKBC}　(Kc)）と、コンテンツを利用するのに必要なライセンスを識別するためのライセンスIDを付加する。そして、ステップＳ２６において、コンテンツサーバ３のCPU２１は、ステップＳ２４の処理で暗号化したコンテンツと、ステップＳ２５の処理でキーとライセンスIDを付加したヘッダとをフォーマット化したデータを、通信部２９から、インターネット２を介して、アクセスしてきたクライアント１に送信する。

【００４０】図５は、このようにして、コンテンツサーバ３からクライアント１にコンテンツが供給される場合のフォーマットの構成を表している。同図に示されるように、このフォーマットは、ヘッダ（Header）とデータ（Data）とにより構成される。

【００４１】ヘッダには、コンテンツ情報（Content information）、デジタル権利管理情報（DRM (Digital Right Management) information）、ライセンスID（License ID）、イネーブリングキーブロック（有効化キーブロック）（EKB（EnablingKey Block））および、EKBから生成されたキーK_{EKBC}を用いて暗号化されたコンテンツキーKcとしてのデータK_{EKBC}　(Kc) が配置されている。なお、EKBについては、図１５を参照して後述する。

【００４２】コンテンツ情報には、データとしてフォーマット化されているコンテンツデータを識別するための識別情報としてのコンテンツID（CID）、そのコンテンツのコーデックの方式などの情報が含まれている。

【００４３】デジタル権利管理情報には、コンテンツを使用する規則および状態（Usage rules/status）と、URL（Uniform Resource Locator）が配置されている。使用規則および状態には、例えば、コンテンツの再生回数、コピー回数などが記述される。

【００４４】URLは、ライセンスIDで規定されるライセンスを取得するときアクセスするアドレス情報であり、図１のシステムの場合、具体的には、ライセンスを受けるために必要なライセンスサーバ４のアドレスである。ライセンスIDは、データとして記録されているコンテンツを利用するとき必要とされるライセンスを識別するものである。

【００４５】データは、任意の数の暗号化ブロック（Encryption Block）により構成される。各暗号化ブロックは、イニシャルベクトル（IV（Initial Vector））、シード（Seed）、およびコンテンツデータをキーK'cで暗号化したデータ$E_{K'c}$ (data)により構成されている。

【００４６】キーK'cは、次式により示されるように、コンテンツキーKcと、乱数で設定される値Seedをハッシュ関数に適用して演算された値により構成される。

【００４７】K'c＝Hash(Kc, Seed)

【００４８】イニシャルベクトルIVとシードSeedは、各暗号化ブロック毎に異なる値に設定される。

【００４９】この暗号化は、コンテンツのデータを８バイト単位で区分して、８バイト毎に行われる。後段の８バイトの暗号化は、前段の８バイトの暗号化の結果を利用して行われるCBC（Cipher Block Chaining）モードで行われる。

【００５０】CBCモードの場合、最初の８バイトのコンテンツデータを暗号化するとき、その前段の８バイトの暗号化結果が存在しないため、最初の８バイトのコンテンツデータを暗号化するときは、イニシャルベクトルIVを初期値として暗号化が行われる。

【００５１】このCBCモードによる暗号化を行うことで、１つの暗号化ブロックが解読されたとしても、その影響が、他の暗号化ブロックにおよぶことが抑制される。

【００５２】なお、この暗号化については、図４６を参照にして、後に詳述する。

【００５３】また、暗号方式についてはこれに限らない。

【００５４】以上のようにして、クライアント１は、コンテンツサーバ３からコンテンツを無料で、自由に取得することができる。従って、コンテンツそのものは、大量に、配布することが可能となる。

【００５５】しかしながら、各クライアント１は、取得したコンテンツを利用するとき、ライセンスを保持している必要がある。そこで、図６を参照して、クライアント１がコンテンツを再生する場合の処理について説明する。

【００５６】ステップＳ４１において、クライアント１のCPU２１は、ユーザが入力部２６を操作することで指示したコンテンツの識別情報（CID）を取得する。この識別情報は、例えば、コンテンツのタイトルや、記憶されている各コンテンツ毎に付与されている番号などにより構成される。

【００５７】そして、CPU２１は、コンテンツが指示されると、そのコンテンツに対応するライセンスID（そのコンテンツを使用するのに必要なライセンスのID）を読み取る。このライセンスIDは、図５に示されるように、暗号化されているコンテンツデータのヘッダに記述されているものである。

【００５８】次に、ステップＳ４２に進み、CPU２１は、ステップＳ４１で読み取られたライセンスIDに対応するライセンスが、クライアント１により既に取得され、記憶部２８に記憶されているか否かを判定する。まだ、ライセンスが取得されていない場合には、ステップＳ４３に進み、CPU２１は、ライセンス取得処理を実行する。このライセンス取得処理の詳細は、図７のフローチャートを参照して後述する。

【００５９】ステップＳ４２において、ライセンスが既に取得されていると判定された場合、または、ステップＳ４３において、ライセンス取得処理が実行された結果、ライセンスが取得された場合、ステップＳ４４に進み、CPU２１は、取得されているライセンスは有効期限内のものであるか否かを判定する。ライセンスが有効期限内のものであるか否かは、ライセンスの内容として規定されている期限（後述する図８参照）と、タイマ２０により計時されている現在日時と比較することで判断される。ライセンスの有効期限が既に満了していると判定された場合、CPU２１は、ステップＳ４５に進み、ライセンス更新処理を実行する。このライセンス更新処理の詳細は、図１０のフローチャートを参照して後述する。

【００６０】ステップＳ４４において、ライセンスはまだ有効期限内であると判定された場合、または、ステップＳ４５において、ライセンスが更新された場合、ステップＳ４６に進み、CPU２１は、暗号化されているコンテンツデータを記憶部２８から読み出し、RAM２３に格納させる。そして、ステップＳ４７において、CPU２１は、RAM２３に記憶された暗号化ブロックのデータを、図５のデータに配置されている暗号化ブロック単位で、暗号化復号部２４に供給し、コンテンツキーKcを用いて復号させる。

【００６１】コンテンツキーKcを得る方法の具体例は、図１５を参照して後述するが、デバイスノードキー（DNK）（図８）を用いて、EKB（図５）に含まれるキーK_{EKBC}を得ることができ、そのキーK_{EKBC}を用いて、データK_{EKBC}（Kc）（図５）から、コンテンツキーKcを得ることができる。

【００６２】CPU２１は、さらに、ステップＳ４８において、暗号化復号部２４により復号されたコンテンツデータをコーデック部２５に供給し、デコードさせる。そして、コーデック部２５によりデコードされたデータを、CPU２１は、入出力インタフェース３２から出力部２７に供給し、D/A変換させ、スピーカから出力させる。

【００６３】次に、図７のフローチャートを参照して、図６のステップＳ４３で行われるライセンス取得処理の詳細について説明する。

【００６４】クライアント１は、事前にライセンスサーバに登録することにより、リーフID、DNK(Device Node Key)、クライアント１の秘密鍵・公開鍵のペア、ライセンスサーバの公開鍵、及び各公開鍵の証明書を含むサービスデータを取得しておく。

【００６５】リーフIDは、クライアント毎に割り当てられた識別情報を表し、DNKは、そのライセンスに対応するEKB（有効化キーブロック）に含まれる暗号化されているコンテンツキーKcを復号するのに必要なデバイスノードキーである（図１２を参照して後述する）。

【００６６】最初にステップＳ６１において、CPU２１は、いま処理対象とされているライセンスIDに対応するURLを、図５に示すヘッダから取得する。上述したように、このURLは、やはりヘッダに記述されているライセンスIDに対応するライセンスを取得するときアクセスすべきアドレスである。そこで、ステップＳ６２において、CPU２１は、ステップＳ６１で取得したURLにアクセスする。具体的には、通信部２９によりインターネット２を介してライセンスサーバ４にアクセスが行われる。このとき、ライセンスサーバ４は、クライアント１に対して、購入するライセンス（コンテンツを使用するのに必要なライセンス）を指定するライセンス指定情報、並びにユーザIDとパスワードの入力を要求してくる（後述する図９のステップＳ１０２）。CPU２１は、この要求を出力部２７の表示部に表示させる。ユーザは、この表示に基づいて、入力部２６を操作して、ライセンス指定情報、ユーザID、およびパスワードを入力する。なお、このユーザIDとパスワードは、クライアント１のユーザが、インターネット２を介してライセンスサーバ４にアクセスし、事前に取得しておいたものである。

【００６７】CPU２１は、ステップＳ６３，Ｓ６４において、入力部２６から入力されたライセンス識別情報を

取り込むとともに、ユーザIDとパスワードを取り込む。CPU２１は、ステップＳ６５において、通信部２９を制御し、入力されたユーザIDとパスワードを、ライセンス指定情報及びサービスデータ（後述する）に含まれるリーフIDを含むライセンス要求をインターネット２を介してライセンスサーバ４に送信させる。

【００６８】ライセンスサーバ４は、図９を参照して後述するように、ユーザIDとパスワード、並びにライセンス指定情報に基づいてライセンスを送信してくる（ステップＳ１０９）か、または、条件が満たされない場合には、ライセンスを送信してこない（ステップＳ１１２）。

【００６９】ステップＳ６６において、CPU２１は、ライセンスサーバ４からライセンスが送信されてきたか否かを判定し、ライセンスが送信されてきた場合には、ステップＳ６７に進み、そのライセンスを記憶部２８に供給し、記憶させる。

【００７０】ステップＳ６６において、ライセンスが送信されて来ないと判定した場合、CPU２１は、ステップＳ６８に進み、エラー処理を実行する。具体的には、CPU２１は、コンテンツを利用するためのライセンスが得られないので、コンテンツの再生処理を禁止する。

【００７１】以上のようにして、各クライアント１は、コンテンツデータに付随しているライセンスIDに対応するライセンスを取得して、初めて、そのコンテンツを使用することが可能となる。

【００７２】なお、図７のライセンス取得処理は、各ユーザがコンテンツを取得する前に、予め行っておくようにすることも可能である。

【００７３】クライアント１に提供されるライセンスは、例えば、図８に示されるように、使用条件、リーフIDおよびを含んでいる。

【００７４】使用条件には、そのライセンスに基づいて、コンテンツを使用することが可能な使用期限、そのライセンスに基づいて、コンテンツをダウンロードすることが可能なダウンロード期限、そのライセンスに基づいて、コンテンツをコピーすることが可能な回数（許されるコピー回数）、チェックアウト回数、最大チェックアウト回数、そのライセンスに基づいて、コンテンツをCD-Rに記録することができる権利、PD（Portable Device）にコピーすることが可能な回数、ライセンスを所有権（買い取り状態）に移行できる権利、使用ログをとる義務等を示す情報が含まれる。

【００７５】次に、図９のフローチャートを参照して、図７のクライアント１のライセンス取得処理に対応して実行されるライセンスサーバ４のライセンス提供処理について説明する。なお、この場合においても、図２のクライアント１の構成は、ライセンスサーバ４の構成として引用される。

【００７６】ステップＳ１０１において、ライセンスサーバ４のCPU２１は、クライアント１よりアクセスを受けるまで待機し、アクセスを受けたとき、ステップＳ１０２に進み、アクセスしてきたクライアント１に対して、ユーザIDとパスワード、並びに、ライセンス指定情報の送信を要求する。上述したようにして、クライアント１から、図７のステップＳ６５の処理で、ユーザIDとパスワード、リーフID並びにライセンス指定情報（ライセンスID）が送信されてきたとき、ライセンスサーバ４のCPU２１は、通信部２９を介してこれを受信し、取り込む処理を実行する。

【００７７】そして、ライセンスサーバ４のCPU２１は、ステップＳ１０３において、通信部２９から課金サーバ５にアクセスし、ユーザIDとパスワードに対応するユーザの与信処理を要求する。課金サーバ５は、インターネット２を介してライセンスサーバ４から与信処理の要求を受けると、そのユーザIDとパスワードに対応するユーザの過去の支払い履歴などを調査し、そのユーザが、過去にライセンスの対価の不払いの実績があるか否かなどを調べ、そのような実績がない場合には、ライセンスの付与を許容する与信結果を送信し、不払いの実績などがある場合には、ライセンス付与の不許可の与信結果を送信する。

【００７８】ステップＳ１０４において、ライセンスサーバ４のCPU２１は、課金サーバ５からの与信結果が、ライセンスを付与することを許容する与信結果であるか否かを判定し、ライセンスの付与が許容されている場合には、ステップＳ１０５に進み、ステップＳ１０２の処理で取り込まれたライセンス指定情報に対応するライセンスを、記憶部２８に記憶されているライセンスの中から取り出す。記憶部２８に記憶されているライセンスは、あらかじめライセンスID、バージョン、作成日時、有効期限等の情報が記述されている。ステップＳ１０６において、CPU２１は、そのライセンスに受信したリーフIDを付加する。さらに、ステップＳ１０７において、CPU２１は、ステップＳ１０５で選択されたライセンスに対応づけられている使用条件を選択する。あるいはまた、ステップＳ１０２の処理で、ユーザから使用条件が指定された場合には、その使用条件が必要に応じて、予め用意されている使用条件に付加される。CPU２１は、選択された使用条件をライセンスに付加する。

【００７９】ステップＳ１０８において、ＣＰＵ２１はライセンスサーバの秘密鍵によりライセンスに署名し、これにより、図８に示されるような構成のライセンスが生成される。

【００８０】次に、ステップＳ１０９に進み、ライセンスサーバ４のCPU２１は、そのライセンス（図８に示される構成を有する）を、通信部２９からインターネット２を介してクライアント１に送信させる。

【００８１】ステップＳ１１０においてライセンスサーバ４のCPU２１は、ステップＳ１０９の処理で、いま送

信したライセンス（使用条件、リーフIDを含む）を、ステップＳ１０２の処理で取り込まれたユーザIDとパスワードに対応して、記憶部２８に記憶させる。さらに、ステップＳ１１１において、CPU２１は、課金処理を実行する。具体的には、CPU２１は、通信部２９から課金サーバ５に、そのユーザIDとパスワードに対応するユーザに対する課金処理を要求する。課金サーバ５は、この課金の要求に基づいて、そのユーザに対する課金処理を実行する。上述したように、この課金処理に対して、そのユーザが支払いを行わなかったような場合には、以後、そのユーザは、ライセンスの付与を要求したとしても、ライセンスを受けることができないことになる。

【００８２】すなわち、この場合には、課金サーバ５からライセンスの付与を不許可とする与信結果が送信されてくるので、ステップＳ１０４からステップＳ１１２に進み、CPU２１は、エラー処理を実行する。具体的には、ライセンスサーバ４のCPU２１は、通信部２９を制御してアクセスしてきたクライアント１に対して、ライセンスを付与することができない旨のメッセージを出力し、処理を終了させる。

【００８３】この場合、上述したように、そのクライアント１はライセンスを受けることができないので、そのコンテンツを利用すること（暗号を復号すること）ができないことになる。

【００８４】図１０は、図６のステップＳ４５におけるライセンス更新処理の詳細を表している。図１０のステップＳ１３１乃至ステップＳ１３５の処理は、図７のステップＳ６１乃至ステップＳ６５の処理と基本的に同様の処理である。ただし、ステップＳ１３３において、CPU２１は、購入するライセンスではなく、更新するライセンスのライセンスIDを取り込む。そして、ステップＳ１３５において、CPU２１は、ユーザIDとパスワードとともに、更新するライセンスのライセンスIDを、ライセンスサーバ４に送信する。

【００８５】ステップＳ１３５の送信処理に対応して、ライセンスサーバ４は、後述するように、使用条件を提示してくる（図１１のステップＳ１５３）。そこで、クライアント１のCPU２１は、ステップＳ１３６において、ライセンスサーバ４からの使用条件の提示を受信し、これを出力部２７に出力し、表示させる。ユーザは、入力部２６を操作して、この使用条件の中から所定の使用条件を選択したり、所定の使用条件を新たに追加したりする。ステップＳ１３７でCPU２１は、以上のようにして選択された使用条件（ライセンスを更新する条件）を購入するための申し込みをライセンスサーバ４に送信する。この申し込みに対応して、後述するようにライセンスサーバ４は、最終的な使用条件を送信してくる（図１１のステップＳ１５４）。そこで、ステップＳ１３８において、クライアント１のCPU２１は、ライセンスサーバ４からの使用条件を取得し、ステップＳ１３９において、その使用条件を記憶部２８にすでに記憶されている対応するライセンスの使用条件として更新する。

【００８６】図１１は、以上のクライアント１のライセンス更新処理に対応して、ライセンスサーバ４が実行するライセンス更新処理を表している。

【００８７】最初に、ステップＳ１５１において、ライセンスサーバ４のCPU２１は、クライアント１からのアクセスを受けると、ステップＳ１５２において、クライアント１がステップＳ１３５で送信したライセンス指定情報をライセンス更新要求情報とともに受信する。

【００８８】ステップＳ１５３において、CPU２１は、ライセンスの更新要求を受信すると、そのライセンスに対応する使用条件（更新する使用条件）を、記憶部２８から読み出し、クライアント１に送信する。

【００８９】この提示に対して、上述したように、クライアント１から使用条件の購入が図１０のステップＳ１３７の処理で申し込まれると、ステップＳ１５４において、ライセンスサーバ４のCPU２１は、申し込まれた使用条件に対応するデータを生成し、ステップＳ１５４において、クライアントと１に送信する。クライアント１は、上述したように、ステップＳ１３９の処理で受信した使用条件を用いて、すでに登録されているライセンスの使用条件を更新する。

【００９０】本発明においては、図１２に示されるように、ブロードキャストインクリプション（Broadcast Encryption）方式の原理に基づいて、デバイスとライセンスのキーが管理される。キーは、階層ツリー構造とされ、最下段のリーフ（leaf）が個々のデバイスのキーに対応する。図１２の例の場合、番号０から番号１５までの１６個のデバイスまたはライセンスに対応するキーが生成される。

【００９１】各キーは、図中丸印で示されるツリー構造の各ノードに対応して規定される。この例では、最上段のルートノードに対応してルートキーKRが、２段目のノードに対応してキーＫ０，Ｋ１が、３段目のノードに対応してキーＫ００乃至Ｋ１１が、第４段目のノードに対応してキーＫ０００乃至キーＫ１１１が、それぞれ対応されている。そして、最下段のノードとしてのリーフ（デバイスノード）に、キーＫ００００乃至Ｋ１１１１が、それぞれ対応されている。

【００９２】階層構造とされているため、例えば、キーＫ００１０とキー００１１の上位のキーは、Ｋ００１とされ、キーＫ０００とキーＫ００１の上位のキーは、Ｋ００とされている。以下同様に、キーＫ００とキーＫ０１の上位のキーは、Ｋ０とされ、キーＫ０とキーＫ１の上位のキーは、KRとされている。

【００９３】コンテンツを利用するキーは、最下段のデバイスノード（リーフ）から、最上段のルートノードまでの１つのパスの各ノードに対応するキーで管理される。例えば、番号３のノード（リーフID）に対応するラ

イセンスに基づき、コンテンツを利用するキーは、キーＫ００１１，Ｋ００１，Ｋ００，Ｋ０，ＫＲを含むパスの各キーで管理される。

【００９４】本発明のシステムにおいては、図１３に示されるように、図１２の原理に基づいて構成されるキーシステムで、デバイスのキーとライセンスのキーの管理が行われる。図１３の例では、８＋２４＋３２段のノードがツリー構造とされ、ルートノードから下位の８段までの各ノードにカテゴリが対応される。ここにおけるカテゴリとは、例えばメモリスティックなどの半導体メモリを使用する機器のカテゴリ、デジタル放送を受信する機器のカテゴリといったカテゴリを意味する。そして、このカテゴリノードのうちの１つのノードに、ライセンスを管理するシステムとして本システム（Ｔシステムと称する）が対応する。

【００９５】すなわち、このＴシステムのノードよりさらに下の階層の２４段のノードに対応するキーにより、ライセンスが対応される。この例の場合、これにより、2^{24}（約１６メガ）のライセンスを規定することができる。さらに、最も下側の３２段の階層により、2^{32}（約４ギガ）のユーザ（あるいはクライアント１）を規定することができる。最下段の３２段のノードに対応するキーが、DNK（Device Node Key）を構成し、最下段のリーフに対応するIDがリーフIDとされる。

【００９６】各デバイスやライセンスのキーは、６４（＝８＋２４＋３２）段の各ノードで構成されるパスの内の１つに対応される。例えば、コンテンツを暗号化したコンテンツキーは、対応するライセンスに割り当てられたパスを構成するノードに対応するキーを用いて暗号化される。上位の階層のキーは、その直近の下位の階層のキーを用いて暗号化され、ＥＫＢ（図１５を参照して後述する）内に配置される。最下段のＤＮＫは、ＥＫＢ内には配置されず、サービスデータに記述され、ユーザのクライアント１に与えられる。クライアント１は、ライセンスに記述されているＤＮＫを用いて、コンテンツデータとともに配布されるＥＫＢ（図１５）内に記述されている直近の上位の階層のキーを復号し、復号して得たキーを用いて、ＥＫＢ内に記述されているさらにその上の階層のキーを復号する。以上の処理を順次行うことで、クライアント１は、そのライセンスのパスに属するすべてのキーを得ることができる。

【００９７】図１４に階層ツリー構造のカテゴリの分類の具体的な例を示す。図１４において、階層ツリー構造の最上段には、ルートキーKR２３０１が設定され、以下の中間段にはノードキー２３０２が設定され、最下段には、リーフキー２３０３が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。

【００９８】最上段から第M段目（図１３の例では、Ｍ＝８）の所定のノードがカテゴリノード２３０４として設定される。すなわち第M段目のノードの各々が特定カテゴリのデバイス設定ノードとされる。第M段の１つのノードを頂点としてＭ＋１段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとされる。

【００９９】例えば図１４の第M段目の１つのノード２３０５にはカテゴリ［メモリスティック（商標）］が設定され、このノード以下に連なるノード、リーフはメモリステッイクを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード２３０５以下が、メモリスティックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義される。

【０１００】さらに、M段から数段分下位の段をサブカテゴリノード２３０６として設定することができる。図１４の例では、カテゴリ［メモリスティック］ノード２３０５の２段下のノードに、メモリスティックを使用したデバイスのカテゴリに含まれるサブカテゴリノードとして、［再生専用器］のノード２３０６が設定されている。さらに、サブカテゴリノードである再生専用器のノード２３０６以下に、再生専用器のカテゴリに含まれる音楽再生機能付き電話のノード２３０７が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる［ＰＨＳ］ノード２３０８と、［携帯電話］ノード２３０９が設定されている。

【０１０１】さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えばあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば１つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器ＸＹＺ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器ＸＹＺに、その頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（ＥＫＢ）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

【０１０２】このように、１つのノードを頂点として、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の１つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（ＥＫＢ）を独自に生成して、頂点ノード以下に属するデバイスに配信する構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには

全く影響を及ぼさずにキー更新を実行することができる。

【０１０３】例えば、図１２に示されるツリー構造において、１つのグループに含まれる４つのデバイス０，１，２，３はノードキーとして共通のキーＫ００、Ｋ０、KRを保有する。このノードキー共有構成を利用することにより、共通のコンテンツキーをデバイス０，１，２，３のみに提供することが可能となる。たとえば、共通に保有するノードキーＫ００自体をコンテンツキーとして設定すれば、新たな鍵送付を実行することなくデバイス０，１，２，３のみが共通のコンテンツキーの設定が可能である。また、新たなコンテンツキーＫｃｏｎをノードキーＫ００で暗号化した値Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を、ネットワークを介してあるいは記録媒体に格納してデバイス０，１，２，３に配布すれば、デバイス０，１，２，３のみが、それぞれのデバイスにおいて保有する共有ノードキーＫ００を用いて暗号Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を解いてコンテンツキーＫｃｏｎを得ることが可能となる。なお、Ｅｎｃ（Ｋａ，Ｋｂ）はＫｂをＫａによって暗号化したデータであることを示す。

【０１０４】また、ある時点ｔにおいて、デバイス３の所有する鍵Ｋ００１１,Ｋ００１,Ｋ００,Ｋ０,KRが攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス０，１，２，３のグループ）で送受信されるデータを守るために、デバイス３をシステムから切り離す必要がある。そのためには、ノードキーＫ００１,Ｋ００,Ｋ０,KRを、それぞれ新たな鍵Ｋ（ｔ）００１,Ｋ（ｔ）００,Ｋ（ｔ）０,Ｋ（ｔ）Ｒに更新し、デバイス０，１，２にその更新キーを伝える必要がある。ここで、Ｋ（ｔ）ａａａは、鍵Ｋａａａの世代（Generation）ｔの更新キーであることを示す。

【０１０５】更新キーの配布処理ついて説明する。キーの更新は、例えば、図１５Ａに示す有効化キーブロック（ＥＫＢ：Enabling Key Block）と呼ばれるブロックデータによって構成されるテーブルを、ネットワークを介して、あるいは記録媒体に格納してデバイス０，１，２に供給することによって実行される。なお、有効化キーブロック（ＥＫＢ）は、図１２に示されるようなツリー構造を構成する各リーフ（最下段のノード）に対応するデバイスに、新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック（ＥＫＢ）は、キー更新ブロック（ＫＲＢ：Key Renewal Block）と呼ばれることもある。

【０１０６】図１５Ａに示す有効化キーブロック（ＥＫＢ）は、ノードキーの更新の必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。図１５Ａの例は、図１２に示すツリー構造中のデバイス０，１，２において、世代ｔの更新ノードキーを配布することを目的として形成されたブロックデータである。図１２から明らかなように、デバイス０，デバイス１は、更新ノードキーとしてＫ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒが必要であり、デバイス２は、更新ノードキーとしてＫ（ｔ）００１、Ｋ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒが必要である。

【０１０７】図１５ＡのＥＫＢに示されるように、ＥＫＢには複数の暗号化キーが含まれる。図１５Ａの最下段の暗号化キーは、Ｅｎｃ（Ｋ００１０，Ｋ（ｔ）００１）である。これはデバイス２の持つリーフキーＫ００１０によって暗号化された更新ノードキーＫ（ｔ）００１であり、デバイス２は、自身の持つリーフキーＫ００１０によってこの暗号化キーを復号し、更新ノードキーＫ（ｔ）００１を得ることができる。また、復号により得た更新ノードキーＫ（ｔ）００１を用いて、図１５Ａの下から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００１，Ｋ（ｔ）００）が復号可能となり、更新ノードキーＫ（ｔ）００を得ることができる。

【０１０８】以下順次、図１５Ａの上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号することで、更新ノードキーＫ（ｔ）０が得られ、これを用いて、図１５Ａの上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号することで、更新ルートキーＫ（ｔ）Ｒが得られる。

【０１０９】一方、ノードキーＫ０００は更新する対象に含まれておらず、ノード０，１が、更新ノードキーとして必要なのは、Ｋ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒである。ノード０，１は、デバイスキーＫ０００００，Ｋ０００１を用いて、図１５Ａの上から３段目の暗号化キーＥｎｃ（Ｋ０００，Ｋ（ｔ）００）を復号することで更新ノードキーＫ（ｔ）００を取得し、以下順次、図１５Ａの上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号することで、更新ノードキーＫ（ｔ）０を得、図１５Ａの上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号することで、更新ルートキーＫ（ｔ）Ｒを得る。このようにして、デバイス０，１，２は更新したキーＫ（ｔ）Ｒを得ることができる。

【０１１０】なお、図１５Ａのインデックスは、図の右側の暗号化キーを復号するための復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

【０１１１】図１２に示すツリー構造の上位段のノードキーＫ（ｔ）０,Ｋ（ｔ）Ｒの更新が不要であり、ノードキーＫ００のみの更新処理が必要である場合には、図１５Ｂの有効化キーブロック（ＥＫＢ）を用いることで、更新ノードキーＫ（ｔ）００をデバイス０，１，２に配布することができる。

【０１１２】図１５Ｂに示すＥＫＢは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図１２に点線で示すグループ内のデバイス０，１，２，３がある記録

媒体を用いており、新たな共通のコンテンツキーＫ（ｔ）ｃｏｎが必要であるとする。このとき、デバイス０，１，２，３の共通のノードキーＫ００を更新したＫ（ｔ）００を用いて新たな共通の更新コンテンツキーＫ（ｔ）ｃｏｎを暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）ｃｏｎ）が、図１５Ｂに示されるＥＫＢとともに配布される。この配布により、デバイス４など、その他のグループの機器が復号することができないデータとしての配布が可能となる。

【０１１３】すなわち、デバイス０，１，２はＥＫＢを処理して得たキーＫ（ｔ）００を用いて暗号文を復号すれば、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得ることが可能になる。

【０１１４】図１６に、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得る処理例として、Ｋ（ｔ）００を用いて新たな共通のコンテンツキーＫ（ｔ）ｃｏｎを暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）ｃｏｎ）と、図１５Ｂに示すＥＫＢとを記録媒体を介して受領したデバイス０の処理を示す。すなわちこの例は、ＥＫＢによる暗号化メッセージデータをコンテンツキーＫ（ｔ）ｃｏｎとした例である。

【０１１５】図１６に示すように、デバイス０は、記録媒体に格納されている世代t時点のＥＫＢと、自分があらかじめ格納しているノードキーＫ０００を用いて、上述したと同様のＥＫＢ処理により、ノードキーＫ（ｔ）００を生成する。さらに、デバイス０は、復号した更新ノードキーＫ（ｔ）００を用いて、更新コンテンツキーＫ（ｔ）ｃｏｎを復号して、後にそれを使用するために自分だけが持つリーフキーＫ００００で暗号化して格納する。

【０１１６】図１７に有効化キーブロック（ＥＫＢ）のフォーマット例を示す。バージョン６０１は、有効化キーブロック（ＥＫＢ）のバージョンを示す識別子である。なお、バージョンは、最新のＥＫＢを識別する機能と、コンテンツとの対応関係を示す機能を持つ。デプスは、有効化キーブロック（ＥＫＢ）の配布先のデバイスに対する階層ツリーの階層数を示す。データポインタ６０３は、有効化キーブロック（ＥＫＢ）中のデータ部６０６の位置を示すポインタであり、タグポインタ６０４はタグ部６０７の位置、署名ポインタ６０５は署名６０８の位置を示すポインタである。

【０１１７】データ部６０６は、例えば更新するノードキーを暗号化したデータを格納する。例えば図１６に示すような更新されたノードキーに関する各暗号化キー等を格納する。

【０１１８】タグ部６０７は、データ部６０６に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを図１８を用いて説明する。

【０１１９】図１８では、データとして先に図１５Ａで説明した有効化キーブロック（ＥＫＢ）を送付する例を示している。この時のデータは、図１８Ｂの表に示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノードアドレスとする。この例の場合は、ルートキーの更新キーＫ（ｔ）Ｒが含まれているので、トップノードアドレスはKRとなる。このとき、例えば最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）は、図１８Ａに示す階層ツリーに示す位置Ｐ０に対応する。次の段のデータは、Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）であり、ツリー上では前のデータの左下の位置Ｐ００に対応する。ツリー構造の所定の位置から見て、その下に、データがある場合は、タグが０、ない場合はタグが１に設定される。タグは｛左（Ｌ）タグ，右（Ｒ）タグ｝として設定される。図１８Ｂの最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）に対応する位置P0の左下の位置P00にはデータがあるので、Ｌタグ＝０、右にはデータがないので、Ｒタグ＝１となる。以下、すべてのデータにタグが設定され、図１８Ｃに示すデータ列、およびタグ列が構成される。

【０１２０】タグは、対応するデータＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）が、ツリー構造のどこに位置しているのかを示すために設定されるものである。データ部６０６に格納されるキーデータＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）・・・は、単純に暗号化されたキーの羅列データに過ぎないが、上述したタグによってデータとして格納された暗号化キーのツリー上の位置が判別可能となる。上述したタグを用いずに、先の図１５で説明した構成のように、暗号化データに対応させたノード・インデックスを用いて、例えば、

０：Ｅｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）
００：Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）
０００：Ｅｎｃ（Ｋ（（ｔ）０００，Ｋ（ｔ）００）
・・・のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると、冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が可能となる。

【０１２１】図１７に戻って、ＥＫＢフォーマットについてさらに説明する。署名（Signature）６０８は、有効化キーブロック（ＥＫＢ）を発行した例えば鍵管理センタ（ライセンスサーバ４）、コンテンツロバイダ（コンテンツサーバ３）、決済機関（課金サーバ５）等が実行する電子署名である。ＥＫＢを受領したデバイスは、署名検証によって正当な有効化キーブロック（ＥＫＢ）発行者が発行した有効化キーブロック（ＥＫＢ）であることを確認する。

【０１２２】以上のようにして、ライセンスサーバ４から供給されたライセンスに基づいて、コンテンツサーバ３から供給されたコンテンツを利用する処理をまとめる

と、図１９に示されるようになる。

【０１２３】すなわち、コンテンツサーバ３からクライアント１に対してコンテンツが提供されるとともに、ライセンスサーバ４からクライアント１にライセンスが供給される。コンテンツは、コンテンツキーKcにより、暗号化されており（Enc（Kc，Content））、コンテンツキーKcは、ルートキーKR(EKBから得られるキーであって、図５におけるキーK_{EKBC}に対応する)で暗号化され（Enc（KR，Kc））、EKBとともに、暗号化されたコンテンツに付加されてクライアント１に提供される。

【０１２４】図１９の例におけるEKBには、例えば、図２０に示されるように、DNKで暗号化したルートキーKRが含まれている（Enc（DNK，KR））。従って、クライアント１は、サービスデータに含まれるDNKを利用して、EKBからルートキーKRを得ることができる。さらに、ルートキーKRを用いて、Enc（KR，Kc）からコンテンツキーKcを復号することができ、コンテンツキーKcを用いて、Enc（Kc，Content）からコンテンツを復号することができる。

【０１２５】このように、クライアント１にDNKを個別に割り当てることにより、図１２と図１５を参照して説明した原理に従って、個々のクライアント１のリボーク（revoke）が可能になる。

【０１２６】また、ライセンスリーフIDを付加して配布することにより、クライアント１において、サービスデータとライセンスの対応付けが行われることになり、ライセンスの不正コピーを防止することが可能になる。

【０１２７】また、クライアント用の証明書と秘密鍵をサービスデータとして配信するようにすることで、エンドユーザも、これらを用いて不正コピーを防止可能なコンテンツを作成することが可能になる。

【０１２８】証明書と秘密鍵の利用については、図２８のフローチャートを参照して後述する。

【０１２９】本発明においては、図１３を参照して説明したように、カテゴリノードにライセンスを管理するTシステムと、各種のコンテンツを利用するデバイスのカテゴリが対応づけられるので、複数のDNKを同一のデバイスに持たせることができる。その結果、異なるカテゴリのコンテンツを１つのデバイスで管理することが可能となる。

【０１３０】図２１は、この関係を表している。すなわち、デバイスＤ１には、Ｔシステムに基づいて、DNK１が割り当てられている、コンテンツ１を利用するライセンスが記録される。同様に、このデバイスＤ１には、例えば、DNK２が割り当てられた、メモリスティックにCDからリッピングしたコンテンツ２を記録することができる。この場合、デバイスＤ１は、コンテンツ１とコンテンツ２という、異なるシステム（Ｔシステムとデバイス管理システム）により配信されたコンテンツを同時に扱うことが可能となる。新たなDNKを割り当てるとき、既に割り当てられているDNKを削除するなどして、デバイスに１個のDNKだけを対応させるようにした場合、このようなことはできない。

【０１３１】また、図１３における、例えば、下側の３２階層の各三角形の１つ１つに、図２２に示されるライセンスカテゴリ１とライセンスカテゴリ２を割り当てることにより、同一のカテゴリ内を、サブカテゴリを利用して、コンテンツのジャンル、レーベル、販売店、配信サービス等の小さな集まりに分類して、管理することが可能となる。

【０１３２】図２２の例においては、例えば、ライセンスカテゴリ１は、ジャズのジャンルに属し、ライセンスカテゴリ２は、ロックのジャンルに属する。ライセンスカテゴリ１には、ライセンスIDが１であるコンテンツ１とコンテンツ２を対応させ、それぞれユーザ１乃至ユーザ３に配布されている。ライセンスカテゴリ２は、ライセンスID２のコンテンツ３、コンテンツ４、およびコンテンツ５が含まれ、それぞれユーザ１とユーザ３に提供されている。

【０１３３】このように、本発明においては、カテゴリ毎に独立したキー管理が可能になる。

【０１３４】また、DNKを、機器やメディアに予め埋め込むのではなく、ライセンスサーバ４により、登録処理を行う際に、各機器やメディアにダウンロードするようにすることで、ユーザによるキーの購入が可能なシステムを実現することができる。

【０１３５】コンテンツは、それが作成された後、どのような使われ方をされようとも、その使われ方に関わりなく、全ての用途において、使用可能であるのが望ましい。例えば、異なるコンテンツ配信サービズ、あるいは使用条件が異なるドメインにおいても、同一のコンテンツが使えることが望ましい。本発明においては、このため、上述したように、各ユーザ（クライアント１）に、認証局としてのライセンスサーバ４から秘密鍵と、それに対応する公開鍵の証明書（certificates）が配布される。各ユーザは、その秘密鍵を用いて、署名（signature）を作成し、コンテンツに付加して、コンテンツの真正さ(integrity)を保証し、かつコンテンツの改竄防止を図ることができる。

【０１３６】この場合の処理の例について、図２３のフローチャートを参照して説明する。図２３の処理は、ユーザがCDから再生したデータを記憶部２８に記憶させるリッピング処理を説明するものである。

【０１３７】最初に、ステップＳ１７１において、クライアント１のCPU２１は、通信部２９を介して入力されるCDの再生データを記録データとして取り込む。ステップＳ１７２において、CPU２１は、ステップＳ１７１の処理で取り込まれた記録データにウォーターマークが含まれているか否かを判定する。このウォーターマークは、３ビットのコピー管理情報（CCI）と、１ビットの

トリガ（Trigger）とにより構成され、コンテンツのデータの中に埋め込まれている。CPU２１は、ウォーターマークが検出された場合には、ステップＳ１７３に進み、そのウォーターマークを抽出する処理を実行する。ウォーターマークが存在しない場合には、ステップＳ１７３の処理はスキップされる。

【０１３８】次に、ステップＳ１７４において、CPU２１は、コンテンツに対応して記録するヘッダのデータを作成する。このヘッダのデータは、コンテンツID、ライセンスID、ライセンスを取得するためのアクセス先を表すURL、およびウォーターマークにより構成される。

【０１３９】次に、ステップＳ１７５に進み、CPU２１は、ステップＳ１７４の処理で作成したヘッダのデータに基づいたデジタル署名を、自分自身の秘密鍵を用いて作成する。この秘密鍵は、ライセンスサーバ４から取得したものである（図７のステップＳ６７）。

【０１４０】ステップＳ１７６で、CPU２１は、暗号化復号部２４を制御し、コンテンツキーでコンテンツを暗号化させる。コンテンツキーは、コンテンツを取得したとき、同時に取得されたものである（図５または図１９）。

【０１４１】次に、ステップＳ１７７において、CPU２１は、ファイルフォーマットに基づき、データを、例えば、ミニディスク等により構成される光磁気ディスク４３に記録させる。

【０１４２】なお、記録媒体がミニディスクである場合、ステップＳ１７６において、CPU２１は、コンテンツをコーデック部２５に供給し、例えば、ATRAC３方式によりコンテンツを符号化させる。そして、符号化されたデータが暗号化復号部２４によりさらに暗号化される。

【０１４３】図２４は、以上のようにして、記録媒体にコンテンツが記録された状態を模式的に表している。暗号化されているコンテンツ（E（At３））から抽出されたウォーターマーク（WM）が、コンテンツの外（ヘッダ）に記録されている。

【０１４４】図２５は、コンテンツを記録媒体に記録する場合のファイルフォーマットのより詳細な構成を表している。この例においては、コンテンツID（CID）、ライセンスID（LID）、URL、およびウォーターマーク（WM）を含むヘッダが記録されている他、EKB、コンテンツキーKcをルートキーKRで暗号化したデータ（Enc（KR, Kc））、証明書（Cert）、ヘッダに基づき生成されたデジタル署名（Sig（Header））、コンテンツをコンテンツキーKcで暗号化したデータ（Enc（Kc, Content））、メタデータ（Meta Data）およびマーク(Mark)が記録されている。

【０１４５】ウォーターマークは、コンテンツの内部に埋め込まれているものであるが、図２４と図２５に示されるように、コンテンツの内部とは別に、ヘッダ内に配置するようにすることで、ウォーターマークとしてコンテンツに埋め込まれている情報を迅速に、かつ簡単に検出することが可能となる。従って、そのコンテンツを、コピーすることができるか否かを、迅速に判定することができる。

【０１４６】なお、メタデータは、例えば、ジャケット、写真、歌詞等のデータを表す。マークについては、図３１を参照して後述する。

【０１４７】図２６は、証明書としての公開鍵証明書の例を表している。公開鍵証明書は、通常、公開鍵暗号方式における認証局（CA：Certificate Authority）が発行する証明書であり、ユーザが、認証局に提出した自己のIDや公開鍵などに、認証局が有効期限等の情報を付加し、さらに、認証局によるデジタル署名を付加して作成される。この発明においては、ライセンスサーバ４（またはコンテンツサーバ３）が、証明書と秘密鍵、従って公開鍵も発行するので、ユーザは、ユーザID、パスワード等をライセンスサーバ４に提供し登録処理を行うことによって、この公開鍵証明書を得ることができる。

【０１４８】図２６における公開鍵証明書は、証明書のバージョン番号、ライセンスサーバ４が証明書の利用者（ユーザ）に対して割りつける証明書の通し番号、デジタル署名に用いたアルゴリズム、およびパラメータ、認証局（ライセンスサーバ４）の名前、証明書の有効期限、証明書利用者のID（ノードIDまたはリーフID）、並びに証明書利用者の公開鍵が、メッセージとして含まれている。さらに、このメッセージには、認証局としてのライセンスサーバ４により作成されたデジタル署名が付加されている。このデジタル署名は、メッセージに対してハッシュ関数を適用して生成されたハッシュ値に基づいて、ライセンスサーバ４の秘密鍵を用いて生成されたデータである。

【０１４９】ノードIDまたはリーフIDは、例えば、図１２の例の場合、デバイス０であれば「００００」とされ、デバイス１であれば「０００１」とされ、デバイス１５であれば「１１１１」とされる。このようなIDに基づいて、そのデバイス（エンティティ）がツリー構成のどの位置（リーフまたはノード）に位置するエンティティであるのかが識別される。

【０１５０】このように、コンテンツを利用するのに必要なライセンスを、コンテンツとは分離して配布するようにすることにより、コンテンツの配布が自由に行われることになる。任意の方法、あるいは経路で入手されたコンテンツは、一元的に取り扱うことが可能である。

【０１５１】また、ファイルフォーマットを図２５に示されるように構成することで、そのフォーマットのコンテンツを、インターネットを介して配信する場合は勿論、SDMI（Secure Digital Music Initiative）機器に提供する場合においても、コンテンツの著作権を管理することが可能となる。

【０１５２】さらに、例えば、図２７に示されるように、コンテンツが記録媒体を介して提供されたとしても、インターネット２を介して提供されたとしても、同様の処理により、SDMI（Secure Digital Music Initiative）機器としての所定のPD（Portable Device）等に、チェックアウトしたりすることが可能となる。

【０１５３】次に、図２８のフローチャートを参照して、クライアント１が他のクライアント（例えば、PD）に対してコンテンツをチェックアウトする場合の処理について説明する。

【０１５４】最初に、ステップＳ１９１において、CPU２１は、コンテンツにデジタル署名が付加されているか否かを判定する。デジタル署名が付加されていると判定された場合、ステップＳ１９２に進み、CPU２１は、証明書を抽出し、認証局（ライセンスサーバ４）の公開鍵で検証する処理を実行する。すなわち、クライアント１は、ライセンスサーバ４からライセンスサーバ４の秘密鍵に対応する公開鍵を取得し、その公開鍵で公開鍵証明書に付加されているデジタル署名を復号する。図２６を参照して説明したように、デジタル署名は、認証局（ライセンスサーバ４）の秘密鍵に基づいて生成されており、ライセンスサーバ４の公開鍵を用いて復号することができる。さらに、CPU２１は、証明書のメッセージ全体に対してハッシュ関数を適用してハッシュ値を演算する。そしてCPU２１は、演算されたハッシュ値と、デジタル署名を復号して得られたハッシュ値とを比較し、両者が一致すれば、メッセージは改竄されたものではないと判定する。両者が一致しない場合には、この証明書は、改竄されたものであるということになる。

【０１５５】そこで、ステップＳ１９３において、CPU２１は、証明書が改竄されていないか否かを判定し、改竄されていないと判定された場合、ステップＳ１９４に進み、証明書をEKBで検証する処理を実行する。この検証処理は、証明書に含まれるリーフID（図２６）に基づいて、EKBをたどることができるか否かを調べることにより行われる。この検証について、図２９と図３０を参照して説明する。

【０１５６】いま、図２９に示されるように、例えば、リーフキーＫ１００１を有するデバイスがリボークされたデバイスであるとする。このとき、図３０に示されるようなデータ（暗号化キー）とタグを有するEKBが、各デバイス（リーフ）に配布される。このEKBは、図２９におけるデバイス「１００１」をリボークするために、キーKR,Ｋ１,Ｋ１０,Ｋ１００を更新するEKBとなっている。

【０１５７】リボークデバイス「１００１」以外の全てのリーフは、更新されたルートキーＫ（ｔ）Ｒを取得することができる。すなわち、ノードキーＫ０の下位に連なるリーフは、更新されていないノードキーＫ０を、デバイス内に保持しているので、暗号化キーEnc（Ｋ０，Ｋ（ｔ）Ｒ）を、キーＫ０によって復号することで、更新ルートキーＫ（ｔ）Ｒを取得することができる。

【０１５８】また、ノードキーＫ１１以下のリーフは、更新されていないノードキーＫ１１を用いて、Enc（Ｋ１１，Ｋ（ｔ）１）をノードキーＫ１１によって復号することで、更新ノードキーＫ（ｔ）１を取得することができる。さらに、Enc（Ｋ（ｔ）１，Ｋ（ｔ）Ｒ）をノードキーＫ（ｔ）１によって復号することで、更新ルートキーＫ（ｔ）Ｒを取得することが可能となる。ノードキーＫ１０１の下位リーフについても、同様に更新ルートキーＫ（ｔ）Ｒを取得することが可能である。

【０１５９】さらに、リボークされていないリーフキーＫ１０００を有するデバイス「１０００」は、自己のリーフキーＫ１０００でEnc（Ｋ１０００，Ｋ（ｔ）１００）を復号して、ノードキーＫ（ｔ）１００を取得することができ、これを用いてさらに、上位のノードキーを順次復号し、更新ルートキーＫ（ｔ）Ｒを取得することができる。

【０１６０】これに対して、リボークされたデバイス「１００１」は、自己のリーフの１段上の更新ノードキーＫ（ｔ）１００を、EKB処理により取得できないので、結局、更新ルートキーＫ（ｔ）Ｒを取得することができない。

【０１６１】リボークされていない正当なデバイス（クライアント１）には、図３０に示されるデータとタグを有するEKBが、ライセンスサーバ４から配信され、格納されている。

【０１６２】そこで、各クライアントは、そのタグを利用して、EKB追跡処理を行うことができる。このEKB追跡処理は、上位のルートキーからキー配信ツリーをたどれるか否かを判定する処理である。

【０１６３】例えば、図２９のリーフ「１００１」のID（リーフID）である「１００１」を、「１」「０」「０」「１」の４ビットとして把握し、最上位ビットから順次、下位ビットに従って、ツリーをたどることができるか否かが判定される。この判定では、ビットが１であれば、右側に進み、０であれば、左側に進む処理が行われる。

【０１６４】ID「１００１」の最上位ビットが１であるから、図２９のルートキーKRから右側に進む。EKBの最初のタグ（番号０のタグ）は、０：｛０，０｝であり、両枝にデータを有するものであると判定される。この場合、右側に進むことができるので、ノードキーＫ１にたどり着くことができる。

【０１６５】次に、ノードキーＫ１の下位のノードに進む。ID「１００１」の２番目のビットは０であるから左側に進む。番号１のタグは、左側のノードキーＫ０の下位のデータの有無を表すものであり、ノードキーＫ１の下位のデータの有無を示すタグは、番号２のタグである。このタグは、図３０に示されるように、２：｛０，

0｝であり、両枝にデータを有するものとされる。従って、左側に進み、ノードキーＫ１０にたどり着くことができる。

【０１６６】さらに、ID「１００１」の３番目のビットは０であり、左側に進む。このとき、Ｋ１０の下位のデータの有無を示すタグ（番号３のタグ）は、３：｛0，0｝であり、両枝にデータを有するものと判定される。そこで、左側に進み、ノードキーＫ１００にたどり着くことができる。

【０１６７】さらに、ID「１００１」の最下位ビットは１であり、右側に進む。番号４のタグは、ノードキーＫ１１に対応するものであり、Ｋ１００の下位のデータの符号を表すタグは、番号５のタグである。このタグは、５：｛0，1｝である。従って、右側には、データが存在しないことになる。その結果、ノード「１００１」にはたどり着けないことになり、ID「１００１」のデバイスは、EKBによる更新ルートキーを取得できないデバイス、すなわちリボークデバイスであると判定される。

【０１６８】これに対して、例えば、リーフキーＫ１００００を有するデバイスIDは、「１０００」であり、上述した場合と同様に、EKB内のタグに基づくEKB追跡処理を行うと、ノード「１０００」にたどり着くことができる。従って、ID「１０００」のデバイスは、正当なデバイスであると判定される。

【０１６９】図２８に戻って、CPU２１は、ステップＳ１９４の検証処理に基づき、証明書がリボークされていないか否かをステップＳ１９５で判定し、証明書がリボークされていない場合には、ステップＳ１９６に進み、デジタル署名を証明書に含まれる公開鍵で検証する処理を実行する。

【０１７０】すなわち、図２６に示されるように、証明書には、証明書利用者（コンテンツ作成者）の公開鍵が含まれており、この公開鍵を用いて、図２５に示される署名（Sig（Header））が検証される。すなわち、この公開鍵を用いて、デジタル署名Sig（Header）を復号して得られたデータ（ハッシュ値）と、図２５に示されるHeaderにハッシュ関数を適用して演算されたハッシュ値とを比較することで、両者が一致していれば、Headerが改竄されていないことを確認することができる。これに対して、両者が一致しなければ、Headerは改竄されているということになる。

【０１７１】ステップＳ１９７において、CPU２１は、Headerが改竄されているか否かを判定し、改竄されていなければ、ステップＳ１９８に進み、ウォーターマークを検証する。ステップＳ１９９において、CPU２１は、ウォーターマークの検証の結果、チェックアウトが可能であるか否かを判定する。チェックアウトが可能である場合には、ステップＳ２００に進み、CPU２１は、チェックアウトを実行する。すなわち、チェックアウト先のクライアント１に対してコンテンツを転送し、コピーさせる。

【０１７２】ステップＳ１９１において、デジタル署名が存在しないと判定された場合、ステップＳ１９３において、証明書が改竄されていると判定された場合、ステップＳ１９５において、証明書をEKBで検証することができなかったと判定された場合、ステップＳ１９７において、デジタル署名の検証の結果、ヘッダが改竄されていると判定された場合、または、ステップＳ１９９において、ウォーターマークにチェックアウトの禁止が記述されていると判定された場合、ステップＳ２０１に進み、エラー処理が実行される。すなわち、この場合には、チェックアウトが禁止される。

【０１７３】このように、証明書と秘密鍵をライセンスサーバ４からユーザに配布し、コンテンツ作成時に、デジタル署名を付加することにより、コンテンツの作成者の真正を保証することが可能となる。これにより、不正なコンテンツの流通を抑制することができる。

【０１７４】さらに、ウォーターマークをコンテンツ作成時に検出し、その情報をデジタル署名に付することで、ウォーターマーク情報の改竄を防止し、コンテンツの真正を保証することができる。

【０１７５】その結果、一度作成されたコンテンツは、どのような形態で配信されたとしても、元のコンテンツの真正を保証することが可能となる。

【０１７６】さらに、コンテンツは、使用条件を有さず、使用条件は、ライセンスに付加されているので、ライセンス内の使用条件を変更することで、それに関係するコンテンツの使用条件を一斉に変更することが可能となる。

【０１７７】次に、マークの利用方法について説明する。本発明においては、上述したように、使用条件は、コンテンツではなく、ライセンスに付加される。しかしながら、コンテンツによって、使用状況が異なる場合がある。そこで、本発明においては、図２５に示されるように、コンテンツにマークが付加される。

【０１７８】ライセンスとコンテンツは、１対多の関係にあるため、コンテンツの個々の使用状況をライセンスの使用条件にのみ記述するのは困難となる。そこで、このように、コンテンツに使用状況を付加することにより、ライセンスでの管理をしながらも、個々のコンテンツを管理することが可能となる。

【０１７９】このマークには、例えば、図３１に示されるように、ユーザのID（リーフID）、所有権フラグ、使用開始時刻、およびコピー回数等が記述される。

【０１８０】さらに、マークには、リーフID、所有権フラグ、使用開始時刻、およびコピー回数等のメッセージに基づいて生成されたデジタル署名が付加される。

【０１８１】所有権フラグは、例えば、所定の期間だけコンテンツを使用可能とするライセンスを、そのまま買い取ったような場合（使用期間を永久に変更したような

場合）に付加される。使用開始時刻は、コンテンツの使用を所定の期間内に開始した場合に記述される。例えば、コンテンツをダウンロードする時期が制限されているような場合において、その期限内にダウンロードが行われたようなとき、その実際にコンテンツをダウンロードした日時がここに記述される。これにより、期間内での有効な使用であることが、証明される。

【０１８２】コピー回数には、それまでにそのコンテンツをコピーした回数が履歴（ログ）として記述される。

【０１８３】次に、図３２のフローチャートを参照して、ユーザがライセンスを買い取った場合に、マークを付加する処理について、マークをコンテンツに付加する例として説明する。

【０１８４】最初に、ステップＳ２２１において、CPU２１は、入力部２６からのユーザの指令に基づいて、インターネット２を介して、ライセンスサーバ４にアクセスする。

【０１８５】ステップＳ２２２において、CPU２１は、ユーザからの入力部２６を介しての入力を取り込み、その入力に対応してライセンスサーバ４に対してライセンスの買い取りを要求する。

【０１８６】この要求に対応して、図３３のフローチャートを参照して後述するように、ライセンスサーバ４は、ライセンスを買い取るために必要な対価を提示してくる（図３３のステップＳ２４２）。そこで、ステップＳ２２３において、クライアント１のCPU２１は、ライセンスサーバ４からの対価の提示を受け取ると、これを出力部２７に出力し、表示させる。

【０１８７】ユーザは、この表示に基づいて、提示された対価を了承するか否かを判断し、その判断結果に基づいて、入力部２６からその判断結果を入力する。

【０１８８】CPU２１は、ステップＳ２２４において、入力部２６からの入力に基づいて、ユーザが提示された対価を了承したか否かを判定し、了承したと判定した場合には、ステップＳ２２５に進み、ライセンスサーバ４に了承を通知する処理を実行する。

【０１８９】この了承通知を受信すると、ライセンスサーバ４は、対価の買い取りを表す情報、すなわち所有権フラグを記述したマークを送信してくる（図３３のステップＳ２４４）。そこで、ステップＳ２２６において、クライアント１のCPU２１は、ライセンスサーバ４からのマークを受け取ると、ステップＳ２２７において、受け取ったマークをコンテンツに埋め込む処理を実行する。すなわち、これにより、買い取られたライセンスに対応するコンテンツのマークとして、図３１に示されるような所有権フラグが記述されたマークがコンテンツに対応して記録されることになる。また、このとき、CPU２１は、メッセージが更新されたことになるので、デジタル署名（図２５）も更新し、記録媒体に記録する。

【０１９０】ステップＳ２２４において、ライセンスサーバ４から提示された対価が了承されていないと判定された場合、ステップＳ２２８に進み、CPU２１は、提示された対価を了承しないことをライセンスサーバ４に通知する。

【０１９１】このようなクライアント１の処理に対応して、ライセンスサーバ４は、図３３のフローチャートに示す処理を実行する。

【０１９２】すなわち、最初に、ステップＳ２４１において、ライセンスサーバ４のCPU２１は、クライアント１からライセンス買い取りの要求が送信されてくると（図３２のステップＳ２２２）、 これを受け取り、ステップＳ２４２において、対象とされているライセンスの買い取りに必要な対価を記憶部２８から読み出し、これをクライアント１に送信する。

【０１９３】上述したように、このようにして提示された対価に対して、クライアント１から提示された対価を了承するか否かの通知が送信されてくる。

【０１９４】そこで、ステップＳ２４３において、ライセンスサーバ４のCPU２１は、クライアント１から了承通知を受信したか否かを判定し、了承通知を受信したと判定した場合、ステップＳ２４４に進み、対象とされるライセンスの買い取りを表すメッセージを含むマークを生成し、自分自身の秘密鍵で、デジタル署名を付加して、クライアント１に送信する。このようにして送信されたマークは、上述したように、クライアント１の記憶部２８において、対応するコンテンツに記録される（図３２のステップＳ２２７）。

【０１９５】ステップＳ２４３において、クライアント１から了承通知が受信されていないと判定された場合、ステップＳ２４４の処理はスキップされる。すなわち、この場合には、ライセンスの買い取り処理が最終的に行われなかったことになるので、マークは送信されない。

【０１９６】図３４は、ステップＳ２４４において、ライセンスサーバ４からクライアント１に対して送信されるマークの構成例を表している。この例においては、そのユーザのリーフID、所有権フラグ（Own）、並びにリーフIDと所有権フラグを、ライセンスサーバ４の秘密鍵Ｓに基づいて生成されたデジタル署名Sig_S（LeafID,Own）により、マークが構成されている。

【０１９７】なお、このマークは、特定のユーザの特定のコンテンツに対してのみ有効なものであるので、対象とされるコンテンツがコピーされた場合には、そのコピーされたコンテンツに付随するマークは無効とされる。

【０１９８】このようにして、コンテンツとライセンスを分離し、使用条件をライセンスに対応させる場合においても、個々のコンテンツの使用状況に応じたサービスを実現することが可能となる。

【０１９９】次に、グルーピングについて説明する。複数の機器やメディアを適当に集め、その１つの集合内においては、コンテンツを自由に授受することができるよ

うにすることは、グルーピングと称される。通常、このグルーピングは、個人の所有する機器やメディアにおいて行われる。このグルーピングは、従来、グループ毎にグループキーを設定する等して行われていたが、グループ化する複数の機器やメディアに、同一のライセンスを対応づけることにより、容易にグルーピングすることが可能となる。

【０２００】また、各機器を予め登録しておくことで、グルーピングすることも可能である。この場合のグルーピングについて、以下に説明する。

【０２０１】この場合、ユーザは、グルーピング対象とされる機器の証明書を予めサーバに登録しておく必要がある。この証明書の登録処理について、図３５と図３６のフローチャートを参照して説明する。

【０２０２】最初に、図３５のフローチャートを参照して、クライアント（グルーピング対象となる機器）の証明書の登録処理について説明する。ステップＳ２６１において、クライアント１のCPU２１は、グルーピングの対象とされる機器としての自分自身の証明書を作成する。この証明書には、自分自身の公開鍵が含まれる。

【０２０３】次に、ステップＳ２６２に進み、CPU２１は、ユーザの入力部２６からの入力に基づいて、コンテンツサーバ３にアクセスし、ステップＳ２６３において、ステップＳ２６１の処理で作成された証明書をコンテンツサーバ３に送信する処理を実行する。

【０２０４】なお、証明書としては、ライセンスサーバ４から受信したものを、そのまま使用することもできる。

【０２０５】以上の処理は、グルーピング対象とされる全ての機器が行う。

【０２０６】次に、図３６のフローチャートを参照して、図３５のクライアント１の証明書の登録処理に対応して行われるコンテンツサーバ３の証明書の登録処理について説明する。

【０２０７】最初に、ステップＳ２７１において、コンテンツサーバ３のCPU２１は、クライアント１から送信されてきた証明書を受信すると、ステップＳ２７２において、その証明書を記憶部２８に登録する。

【０２０８】以上の処理が、グループ対象とされる機器毎に行われる。その結果、コンテンツサーバ３の記憶部２８には、例えば、図３７に示されるように、グループ毎に、そのグループを構成するデバイスの証明書が登録される。

【０２０９】図３７に示される例では、グループ１の証明書として、証明書Ｃ１１乃至Ｃ１４が登録されている。これらの証明書Ｃ１１乃至Ｃ１４には、対応する公開鍵K_{P11}乃至K_{P14}が含まれている。

【０２１０】同様に、グループ２の証明書として、証明書Ｃ２１乃至Ｃ２３が登録されており、これらは対応する公開鍵K_{P21}乃至K_{P23}が含まれている。

【０２１１】以上のようなグループを構成する各機器毎に、その証明書が登録された状態において、ユーザからそのグループに属する機器にコンテンツの提供が要求されると、コンテンツサーバ３は、図３８のフローチャートに示す処理を実行する。

【０２１２】最初に、ステップＳ２８１において、コンテンツサーバ３のCPU２１は、記憶部２８に記憶されている証明書のうち、そのグループに属する証明書を検証する処理を実行する。

【０２１３】この検証処理は、図２９と図３０を参照して説明されたように、各機器の証明書に含まれるリーフIDに基づいて、タグを利用してEKBをたどることで行われる。EKBは、コンテンツサーバ３にも、ライセンスサーバ４から配布されている。この検証処理により、リボークされている証明書は除外される。

【０２１４】ステップＳ２８２において、コンテンツサーバ３のCPU２１は、ステップＳ２８１の検証処理の結果、有効とされた証明書を選択する。そして、ステップＳ２８３において、CPU２１は、ステップＳ２８２の処理で選択された各機器の証明書の各公開鍵でコンテンツ鍵を暗号化する。ステップＳ２８４において、CPU２１は、対象とされるグループの各機器に、ステップＳ２８３の処理で暗号化されたコンテンツ鍵をコンテンツとともに送信する。

【０２１５】図３７に示されるグループ１のうち、例えば、証明書Ｃ１４がリボークされているとすると、ステップＳ２８３の処理で、例えば、図３９に示されるような暗号化データが生成される。

【０２１６】すなわち、図３９の例においては、コンテンツ鍵K_Cが、証明書Ｃ１１の公開鍵K_{P11}、証明書Ｃ１２の公開鍵K_{P12}、または証明書Ｃ１３の公開鍵K_{P13}により、暗号化されている。

【０２１７】コンテンツサーバ３の図３８に示されるような処理に対応して、コンテンツの提供を受ける各グループの機器（クライアント）は、図４０のフローチャートに示す処理を実行する。

【０２１８】最初に、ステップＳ２９１において、クライアント１のCPU２１は、コンテンツサーバ３が図３８のステップＳ２８４の処理で送信してきたコンテンツを、コンテンツ鍵とともに受信する。コンテンツは、コンテンツ鍵K_Cにより、暗号化されており、コンテンツ鍵は上述したように、各機器が保持する公開鍵により暗号化されている（図３９）。

【０２１９】そこで、ステップＳ２９２において、CPU２１は、ステップＳ２９１の処理で受信した自分宛のコンテンツ鍵を、自分自身の秘密鍵で復号し、取得する。そして、取得したコンテンツ鍵を用いてコンテンツの復号処理が行われる。

【０２２０】例えば、図３９の例に示される証明書Ｃ１１に対応する機器は、公開鍵K_{P11}に対応する自分自身

の秘密鍵を用いて、コンテンツ鍵Kcの暗号を復号し、コンテンツ鍵Kcを取得する。そして、コンテンツ鍵Kcを用いて、コンテンツがさらに復号される。

【０２２１】同様の処理は、証明書Ｃ１２，Ｃ１３に対応する機器においても行われる。リボークされている証明書Ｃ１４の機器は、自分自身の公開鍵を用いて暗号化されたコンテンツ鍵Kcがコンテンツに付随して送られてこないので、コンテンツ鍵Kcを復号することができず、従って、コンテンツ鍵Kcを用いてコンテンツを復号することができない。

【０２２２】以上においては、コンテンツキー（すなわちコンテンツ）に対してグルーピングを行うようにしたが、ライセンスキー（ライセンス）に対してグルーピングを行うことも可能である。

【０２２３】以上のようにして、特別なグループキーや、後述するICV（Integrity Check Value）を用いずにグループ化が可能となる。このグループ化は、小規模のグループに適用するのに向いている。

【０２２４】本発明においては、ライセンスもチェックアウト、あるいはチェックインしたり、ムーブしたり、コピーしたりすることが可能とされる。但し、これらの処理はSDMIで定められたルールに基づいて行われる。

【０２２５】次に、図４１と図４２のフローチャートを参照して、このようなクライアントによるライセンスのチェックアウト処理について説明する。

【０２２６】最初に、図４１のフローチャートを参照して他のクライアントにライセンスをチェックアウトするクライアントの処理について説明する。最初に、ステップＳ３０１において、クライアント１のCPU２１は、チェックアウト対象のライセンスのチェックアウト回数Ｎ１を読み取る。このチェックアウト回数は、図８に示される使用条件に書き込まれているので、この使用条件から読み取られる。

【０２２７】次に、ステップＳ３０２において、CPU２１は、チェックアウト対象のライセンスの最大チェックアウト回数Ｎ２を、やはりライセンスの使用条件から読み取る。

【０２２８】そして、ステップＳ３０３において、CPU２１は、ステップＳ３０１の処理で読み取られたチェックアウト回数Ｎ１と、ステップＳ３０２の処理で読み取られた最大チェックアウト回数Ｎ２とを比較し、チェックアウト回数Ｎ１が最大チェックアウト回数Ｎ２より大きいか否かを判定する。

【０２２９】チェックアウト回数Ｎ１が、最大チェックアウト回数Ｎ２より小さいと判定された場合、ステップＳ３０４に進み、CPU２１は、相手側の装置（チェックアウト先のクライアント）のリーフキーを相手個々の装置から取得し、そのリーフキーを、いまチェックアウト対象とされているライセンスIDに対応して記憶部２８のチェックアウトリストに記憶させる。

【０２３０】次に、ステップＳ３０５において、CPU２１は、ステップＳ３０１の処理で読み取られたライセンスのチェックアウト回数Ｎ１の値を１だけインクリメントする。ステップＳ３０６において、CPU２１は、ライセンスのメッセージに基づいて、ICVを演算する。このICVについては、図４６乃至図５０を参照して後述する。ICVを用いてライセンスの改竄を防止することが可能となる。

【０２３１】次に、ステップＳ３０７において、CPU２１は、チェックアウト対象のライセンスと、ステップＳ３０６の処理で演算されたICVを、自分自身の公開鍵を用いて暗号化して、EKBおよび証明書とともに、相手側の装置に出力し、コピーさせる。さらに、ステップＳ３０８において、CPU２１は、ステップＳ３０６の処理で演算されたICVを、相手側装置のリーフキーと、ライセンスIDに対応して記憶部２８のチェックリスト中に記憶させる。

【０２３２】ステップＳ３０３において、チェックアウト回数Ｎ１が最大チェックアウト回数Ｎ２より小さくない（例えば、等しい）と判定された場合、もはや許容される回数だけチェックアウトが行われているので、これ以上チェックアウトを行うことができない。そこで、ステップＳ３０９に進み、CPU２１は、エラー処理を実行する。すなわち、この場合、チェックアウト処理は実行されないことになる。

【０２３３】次に、図４２のフローチャートを参照して、図４１のチェックアウト処理により、ライセンスのチェックアウトを受けるクライアントの処理について説明する。

【０２３４】最初に、ステップＳ３２１において、相手側装置（ライセンスをチェックアウトするクライアント１）に、自分自身のリーフキーを送信する。このリーフキーは、ステップＳ３０４において、相手側のクライアントにより、ライセンスIDに対応して記憶される。

【０２３５】次に、ステップＳ３２２において、CPU２１は、相手側のクライアント１から暗号化されたライセンスとICVが、EKBおよび証明書とともに送信されてきた場合、これを受信する。すなわち、このライセンス、ICV、EKBおよび証明書は、図４１のステップＳ３０７の処理で相手側の装置から送信されたものである。

【０２３６】ステップＳ３２３において、CPU２１は、ステップＳ３２２の処理で受信したライセンス、ICV、EKBおよび証明書を、記憶部２８に記憶させる。

【０２３７】以上のようにして、ライセンスのチェックアウトを受けたクライアント１は、チェックアウトを受けたそのライセンスを使用して、所定のコンテンツを再生する場合、図４３のフローチャートに示される処理を実行する。

【０２３８】すなわち、最初に、ステップＳ３４１において、クライアント１のCPU２１は、ユーザより入力部

２６を介して再生が指定されたコンテンツのICVを演算する。そして、ステップＳ３４２において、CPU２１は、記憶部２８に記憶されている暗号化されているICVを、証明書に含まれている公開鍵に基づいて、復号させる。

【０２３９】次に、ステップＳ３４３において、CPU２１は、ステップＳ３４１の処理により、いま演算されたICVと、ステップＳ３４２の処理により読み出され、復号されたICVが一致するか否かを判定する。両者が一致する場合には、ライセンスは改竄されていないことになる。そこで、ステップＳ３４４にすすみ、CPU２１は、対応するコンテンツを再生する処理を実行する。

【０２４０】これに対して、ステップＳ３４３において、２つのICVが一致しないと判定された場合、ライセンスは改竄されている恐れがある。このため、ステップＳ３４５に進み、CPU２１は、エラー処理を実行する。すなわち、このとき、そのライセンスを用いてコンテンツを再生することができないことになる。

【０２４１】次に、以上のようにして、他のクライアントに一旦チェックアウトしたライセンスのチェックインを受けるクライアントの処理について、図４４のフローチャートを参照して説明する。

【０２４２】最初に、ステップＳ３６１において、CPU２１は、相手側の装置（ライセンスを返却（チェックイン）してくるクライアント１）のリーフキーと、チェックイン対象のライセンスのIDを取得する。次に、ステップＳ３６２において、CPU２１は、ステップＳ３６１で取得されたチェックイン対象のライセンスが、自分自身が相手側装置にチェックアウトしたライセンスであるか否かを判定する。この判定は、図４１のステップＳ３０８の処理で記憶されたICV、リーフキー、およびライセンスIDに基づいて行われる。すなわち、ステップＳ３６１で取得されたリーフキー、ライセンスID、およびICVが、チェックアウトリスト中に記憶されているか否かが判定され、記憶されている場合には、自分自身がチェックアウトしたライセンスであると判定される。

【０２４３】ライセンスが、自分自身がチェックアウトしたものであるとき、ステップＳ３６３において、CPU２１は、相手側の装置のライセンス、EKBおよび証明書の削除を要求する。後述するように、この要求に基づいて、相手側の装置は、ライセンス、EKBおよび証明書の削除を実行する（図４５のステップＳ３８３）。

【０２４４】ステップＳ３６４において、CPU２１は、一旦チェックアウトしたライセンスが再びチェックインされてきたので、そのライセンスのチェックアウト回数Ｎ１を１だけデクリメントする。

【０２４５】ステップＳ３６５において、CPU２１は、相手側の装置に他のライセンスをチェックアウトしているか否かを判定し、まだチェックアウトしている他のライセンスが存在しない場合には、ステップＳ３６６に進み、CPU２１は、相手側の装置のチェックイン対象機器としてのチェックアウトリストにおける記憶を削除する。これに対して、ステップＳ３６５において、相手側の装置にチェックアウトしている他のライセンスが存在すると判定された場合には、他のライセンスのチェックインを受ける可能性があるので、ステップＳ３６６の処理はスキップされる。

【０２４６】ステップＳ３６２において、チェックイン対象とされているライセンスが、自分自身が相手側装置にチェックアウトしたライセンスではないと判定された場合、CPU２１は、ステップＳ３６７に進み、エラー処理を実行する。すなわち、この場合には、自分自身が管轄するライセンスではないことになるので、チェックイン処理は実行されない。

【０２４７】ユーザが、ライセンスを不正にコピーしたような場合、記憶されているICVの値と、ステップＳ３６１の処理で取得されたライセンスに基づいて演算されたICVの値が異なるものとなるで、チェックインできないことになる。

【０２４８】図４５は、図４４のフローチャートに示されるライセンスのチェックイン処理を実行するクライアントに対して、自分自身が有しているライセンスをチェックインさせるクライアントの処理を表している。

【０２４９】ステップＳ３８１において、クライアント１のCPU２１は、相手側の装置（図４４のフローチャートに示す処理を実行するクライアント１）にリーフキーとチェックイン対象のライセンスのIDを送信する。上述したように、相手側の装置は、ステップＳ３６１において、このリーフキーとライセンスIDを取得し、ステップＳ３６２において、それに基づいて、チェックイン対象のライセンスの認証処理を実行する。

【０２５０】ステップＳ３８２において、クライアント１のCPU２１は、相手側の装置からライセンスの削除を要求されたか否かを判定する。すなわち、ライセンスが正当なチェックイン対象のライセンスである場合、上述したように、相手側の装置は、ステップＳ３６３の処理でライセンス、EKBおよび証明書の削除を要求してくる。そこで、この要求を受信した場合、ステップＳ３８３に進み、CPU２１は、ライセンス、EKBおよび証明書を削除する。すなわち、これにより、このクライアント１は、以後そのライセンスを使用できない状態となり、図４４のステップＳ３６４の処理により、チェックアウト回数Ｎ１が、１だけデクリメンドされるので、チェックインが完了したことになる。

【０２５１】ステップＳ３８２において、相手側の装置からライセンスの削除が要求されていないと判定された場合、ステップＳ３８４に進み、エラー処理が実行される。すなわち、この場合には、ICVの値が異なっている等の理由により、チェックインができないことになる。

【０２５２】以上においては、チェックインとチェック

アウトについて説明したが、同様に、ライセンスをコピーあるいはムーブさせるようにすることも可能である。

【０２５３】次に、ライセンス（コンテンツも同様）の改竄を防止するためにライセンスのインテグリティ・チェック値（ＩＣＶ）を生成して、ライセンスに対応付けて、ＩＣＶの計算により、ライセンス改竄の有無を判定する処理構成について説明する。

【０２５４】ライセンスのインテグリティ・チェック値（ＩＣＶ）は、例えばライセンスに対するハッシュ関数を用いて計算され、ＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，Ｌ１，Ｌ２，・・・）によって計算される。ＫｉｃｖはＩＣＶ生成キーである。Ｌ１，Ｌ２はライセンスの情報であり、ライセンスの重要情報のメッセージ認証符号（ＭＡＣ：Message authentication Code）が使用される。

【０２５５】ＤＥＳ暗号処理構成を用いたＭＡＣ値生成例を図４６に示す。図４６の構成に示すように対象となるメッセージを８バイト単位に分割し、（以下、分割されたメッセージをＭ１、Ｍ２、・・・、ＭＮとする）、まず、初期値（IV）とＭ１を、演算部２４－１Ａにより排他的論理和する（その結果をＩ１とする）。次に、Ｉ１をＤＥＳ暗号化部２４－１Ｂに入れ、鍵（以下、Ｋ１とする）を用いて暗号化する（出力をＥ１とする）。続けて、Ｅ１およびＭ２を演算部２４－２Ａにより排他的論理和し、その出力Ｉ２をＤＥＳ暗号化部２４－２Ｂへ入れ、鍵Ｋ１を用いて暗号化する（出力Ｅ２）。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。ＤＥＳ暗号化部２４－NBから最後に出てきたＥＮがメッセージ認証符号（ＭＡＣ（Message Authentication Code））となる。

【０２５６】このようなライセンスのＭＡＣ値とＩＣＶ生成キーにハッシュ関数を適用してライセンスのインテグリティ・チェック値（ＩＣＶ）が生成される。例えばライセンス生成時に生成したＩＣＶと、新たにライセンスに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればライセンスに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。

【０２５７】次に、ライセンスのインテグリティ・チェック値（ＩＣＶ）生成キーであるＫｉｃｖを上述の有効化キーブロックによって送付する構成について説明する。すなわちＥＫＢによる暗号化メッセージデータをライセンスのインテグリティ・チェック値（ＩＣＶ）生成キーとした例である。

【０２５８】図４７および図４８に複数のデバイスに共通のライセンスを送付した場合、それらのライセンスの改竄の有無を検証するためのインテグリティ・チェック値生成キーＫｉｃｖを有効化キーブロック（ＥＫＢ）によって配信する構成例を示す。図４７はデバイス０，１，２，３に対して復号可能なチェック値生成キーＫｉｃｖを配信する例を示し、図４８はデバイス０，１，２，３中のデバイス３をリボーク（排除）してデバイス０，１，２に対してのみ復号可能なチェック値生成キーＫｉｃｖを配信する例を示す。

【０２５９】図４７の例では、更新ノードキーＫ（ｔ）００によって、チェック値生成キーＫｉｃｖを暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）とともに、デバイス０，１，２，３においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキーＫ（ｔ）００を復号可能な有効化キーブロック（ＥＫＢ）を生成して配信する。それぞれのデバイスは、図４７の右側に示すように、まず、ＥＫＢを処理（復号）することにより、更新されたノードキーＫ（ｔ）００を取得し、次に、取得したノードキーＫ（ｔ）００を用いて、暗号化されたチェック値生成キーＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）を復号して、チェック値生成キーＫｉｃｖを得ることが可能となる。

【０２６０】その他のデバイス４，５，６，７・・・は同一の有効化キーブロック（ＥＫＢ）を受信しても自身の保有するノードキー、リーフキーでは、ＥＫＢを処理して更新されたノードキーＫ（ｔ）００を取得することができないので、安全に正当なデバイスに対してのみチェック値生成キーを送付することができる。

【０２６１】一方、図４８の例は、図１２の点線枠で囲んだグループにおいてデバイス３が、例えば鍵の漏洩によりリボーク（排除）されているとして、他のグループのメンバ、すなわち、デバイス０，１，２，に対してのみ復号可能な有効化キーブロック（ＥＫＢ）を生成して配信した例である。図４８に示す有効化キーブロック（ＥＫＢ）と、チェック値生成キー（Ｋｉｃｖ）をノードキー（Ｋ（ｔ）００）で暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）を配信する。

【０２６２】図４８の右側には、復号手順を示してある。デバイス０，１，２は、まず、受領した有効化キーブロックから自身の保有するリーフキーまたはノードキーを用いた復号処理により、更新ノードキー（Ｋ（ｔ）００）を取得する。次に、Ｋ（ｔ）００による復号によりチェック値生成キーＫｉｃｖを取得する。

【０２６３】図１２に示す他のグループのデバイス４，５，６・・・は、この同様のデータ（ＥＫＢ）を受信したとしても、自身の保有するリーフキー、ノードキーを用いて更新ノードキー（Ｋ（ｔ）００）を取得することができない。同様にリボークされたデバイス３においても、自身の保有するリーフキー、ノードキーでは、更新ノードキー（Ｋ（ｔ）００）を取得することができず、正当な権利を有するデバイスのみがチェック値生成キーを復号して利用することが可能となる。

【０２６４】このように、ＥＫＢを利用したチェック値生成キーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能としたチェック値生成キーを配信することが可能となる。

【０２６５】このようなライセンスのインテグリティ・

チェック値（ＩＣＶ）を用いることにより、ＥＫＢと暗号化ライセンスの不正コピーを排除することができる。例えば図４９Ａに示すように、ライセンスＬ１とライセンスＬ２とをそれぞれのライセンスキーを取得可能な有効化キーブロック（ＥＫＢ）とともに格納したメディア１があり、これをそのままメディア２にコピーした場合を想定する。ＥＫＢと暗号化ライセンスのコピーは可能であり、これをＥＫＢを復号可能なデバイスでは利用できることになる。

【０２６６】図４９Ｂに示す例では、各メディアに正当に格納されたライセンスに対応付けてインテグリティ・チェック値（ＩＣＶ（Ｌ１，Ｌ２））を格納する構成とする。なお、（ＩＣＶ（Ｌ１，Ｌ２））は、ライセンスＬ１とライセンスＬ２にハッシュ関数を用いて計算されるライセンスのインテグリティ・チェック値であるＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，Ｌ１，Ｌ２）を示している。図４９Ｂの構成において、メディア１には正当にライセンス１とライセンス２が格納され、ライセンスＬ１とライセンスＬ２に基づいて生成されたインテグリティ・チェック値（ＩＣＶ（Ｌ１，Ｌ２））が格納される。また、メディア２には正当にライセンス１が格納され、ライセンスＬ１に基づいて生成されたインテグリティ・チェック値（ＩＣＶ（Ｌ１））が格納される。

【０２６７】この構成において、メディア１に格納された｛ＥＫＢ，ライセンス２｝をメディア２にコピーしたとすると、メディア２で、ライセンスチェック値を新たに生成すると、ＩＣＶ（Ｌ１，Ｌ２）が生成されることになり、メディア２に格納されているＫｉｃｖ（Ｌ１）と異なり、ライセンスの改竄あるいは不正なコピーによる新たなライセンスの格納が実行されたことが明らかになる。メディアを再生するデバイスにおいて、再生ステップの前ステップにＩＣＶチェックを実行して、生成ＩＣＶと格納ＩＣＶの一致を判別し、一致しない場合は、再生を実行しない構成とすることにより、不正コピーのライセンスの再生を防止することが可能となる。

【０２６８】また、さらに、安全性を高めるため、ライセンスのインテグリティ・チェック値（ＩＣＶ）を書き換えカウンタを含めたデータに基づいて生成する構成としてもよい。すなわちＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，ｃｏｕｎｔｅｒ＋１，Ｌ１，Ｌ２，・・・）によって計算する構成とする。ここで、カウンタ（ｃｏｕｎｔｅｒ＋１）は、ＩＣＶの書き換えごとに１つインクリメントされる値として設定する。なお、カウンタ値はセキュアなメモリに格納する構成とすることが必要である。

【０２６９】さらに、ライセンスのインテグリティ・チェック値（ＩＣＶ）をライセンスと同一メディアに格納することができない構成においては、ライセンスのインテグリティ・チェック値（ＩＣＶ）をライセンスとは別のメディア上に格納する構成としてもよい。

【０２７０】例えば、読み込み専用メディアや通常のＭＯ等のコピー防止策のとられていないメディアにライセンスを格納する場合、同一メディアにインテグリティ・チェック値（ＩＣＶ）を格納するとＩＣＶの書き換えが不正なユーザによりなされる可能性があり、ＩＣＶの安全性が保てないおそれがある。この様な場合、ホストマシン上の安全なメディアにＩＣＶを格納して、ライセンスのコピーコントロール（例えばcheck-in/check-out、move）にＩＣＶを使用する構成とすることにより、ＩＣＶの安全な管理およびライセンスの改竄チェックが可能となる。

【０２７１】この構成例を図５０に示す。図５０では読み込み専用メディアや通常のＭＯ等のコピー防止策のとられていないメディア２２０１にライセンス１乃至ライセンス３が格納され、これらのライセンスに関するインテグリティ・チェック値（ＩＣＶ）を、ユーザが自由にアクセスすることの許可されないホストマシン上の安全なメディア２２０２に格納し、ユーザによる不正なインテグリティ・チェック値（ＩＣＶ）の書き換えを防止した例である。このような構成として、例えばメディア２２０１を装着したデバイスが、メディア２２０１の再生を実行する際にホストマシンであるＰＣ、サーバにおいてＩＣＶのチェックを実行して再生の可否を判定する構成とすれば、不正なコピーライセンスあるいは改竄ライセンスの再生を防止できる。

【０２７２】本発明が適用されるクライアントは、いわゆるパーソナルコンピュータ以外に、PDA（Personal Digital Assistants）、携帯電話機、ゲーム端末機などとすることができる。

【０２７３】一連の処理をソフトウエアにより実行させる場合には、そのソフトウエアを構成するプログラムが、専用のハードウエアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、ネットワークや記録媒体からインストールされる。

【０２７４】この記録媒体は、図２に示されるように、装置本体とは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク４１（フロッピディスクを含む）、光ディスク４２（CD-ROM(Compact Disk - ReadOnly Memory),DVD(Digital Versatile Disk)を含む）、光磁気ディスク４３（ＭＤ（Mini-Disk）を含む）、もしくは半導体メモリ４４などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに提供される、プログラムが記録されているROM２２や、記憶部２８に含まれるハードディスクなどで構成される。

【０２７５】なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に

実行される処理をも含むものである。

【０２７６】また、セキュリティに関連する処理を実行させるプログラムは、その処理を解析されるのを防ぐため、そのプログラム自体が暗号化されているのが望ましい。例えば、暗号処理などを行う処理については、そのプログラムをタンパーレジスタントモジュールとして構成することができる。

【０２７７】また、コンテンツを利用許可するライセンスを特定するためにコンテンツのヘッダに記載されている情報はライセンスを一意に識別するライセンスIDでなくてもよい。上記の実施例では、ライセンスＩＤが、コンテンツの利用に必要なライセンスを特定する情報であり、あるライセンスが利用を許可するコンテンツを特定する情報であり、クライアント１からライセンス要求によって要求されるライセンスを識別する情報である。コンテンツにコンテンツのそのコンテンツに関する各種属性情報のリストが記載され、ライセンスに、そのライセンスによって利用許可されるコンテンツの条件式を記載するようにしても良い。この場合では、コンテンツに含まれる属性情報がそのコンテンツの利用を許可するライセンスを特定する情報であり、ライセンスに含まれる条件式がそのライセンスが利用を許可するコンテンツを特定する情報であり、ライセンスＩＤはライセンスを一意に識別する情報となる。このようにした場合には、一つのコンテンツに複数のライセンスを対応付けることが可能になり、ライセンスの発行を柔軟に行うことができる。

【０２７８】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【０２７９】

【発明の効果】以上の如く、本発明の情報処理装置および方法、ライセンスサーバ、並びにプログラムによれば、暗号化されたデータを自由に配布できるようにし、別途ライセンスを取得することでコンテンツを利用できるようにしたことで、コンテンツの流通を妨げることなく、著作権を保護し、適切な使用料の徴収をすることができる。

【図面の簡単な説明】

【図１】本発明を適用したコンテンツ提供システムの構成を示すブロック図である。

【図２】図１のクライアントの構成を示すブロック図である。

【図３】図１のクライアントのコンテンツのダウンロード処理を説明するフローチャートである。

【図４】図１のコンテンツサーバのコンテンツ提供処理を説明するフローチャートである。

【図５】図４のステップＳ２６におけるフォーマットの例を示す図である。

【図６】図１のクライアントのコンテンツ再生処理を説明するフローチャートである。

【図７】図６のステップＳ４３のライセンス取得処理の詳細を説明するフローチャートである。

【図８】ライセンスの構成を示す図である。

【図９】図１のライセンスサーバのライセンス提供の処理を説明するフローチャートである。

【図１０】図６のステップＳ４５におけるライセンス更新処理の詳細を説明するフローチャートである。

【図１１】図１のライセンスサーバのライセンス更新処理を説明するフローチャートである。

【図１２】キーの構成を説明する図である。

【図１３】カテゴリノードを説明する図である。

【図１４】ノードとデバイスの対応の具体例を示す図である。

【図１５】有効化キーブロックの構成を説明する図である。

【図１６】有効化キーブロックの利用を説明する図である。

【図１７】有効化キーブロックのフォーマットの例を示す図である。

【図１８】有効化キーブロックのタグの構成を説明する図である。

【図１９】DNKを用いたコンテンツの復号処理を説明する図である。

【図２０】有効化キーブロックの例を示す図である。

【図２１】複数のコンテンツの１つのデバイスに対する割り当てを説明する図である。

【図２２】ライセンスのカテゴリを説明する図である。

【図２３】クライアントのリッピング処理を説明するフローチャートである。

【図２４】ウォーターマークの構成を説明する図である。

【図２５】コンテンツのフォーマットの例を示す図である。

【図２６】公開鍵証明書の例を示す図である。

【図２７】コンテンツの配布を説明する図である。

【図２８】クライアントのコンテンツのチェックアウト処理を説明するフローチャートである。

【図２９】タグによる有効化キーブロックをたどる例を説明する図である。

【図３０】有効化キーブロックの構成例を示す図である。

【図３１】マークの構成を説明する図である。

【図３２】クライアントのライセンス買い取り処理を説明するフローチャートである。

【図３３】ライセンスサーバのライセンス買い取り処理を説明するフローチャートである。

【図３４】マークの構成例を示す図である。

【図３５】クライアントの証明書の登録処理を説明するフローチャートである。

【図３６】コンテンツサーバの証明書登録処理を説明するフローチャートである。
【図３７】グループの証明書の例を示す図である。
【図３８】グルーピングが行われている場合におけるコンテンツサーバの処理を説明するフローチャートである。
【図３９】コンテンツキーの暗号化の例を示す図である。
【図４０】グループに属するクライアントの処理を説明するフローチャートである。
【図４１】他のクライアントにライセンスをチェックアウトするクライアントの処理を説明するフローチャートである。
【図４２】他のクライアントからライセンスのチェックアウトを受けるクライアントの処理を説明するフローチャートである。
【図４３】ライセンスのチェックアウトを受けたクライアントの再生処理を説明するフローチャートである。
【図４４】他のクライアントからライセンスのチェックインを受けるクライアントの処理を説明するフローチャートである。
【図４５】他のクライアントにライセンスをチェックインするクライアントの処理を説明するフローチャートである。
【図４６】MACの生成を説明する図である。
【図４７】ICV生成キーの復号処理を説明するフローチャートである。
【図４８】ICV生成キーの他の復号処理を説明する図である。
【図４９】ICVによるライセンスのコピーの管理を説明する図である。
【図５０】ライセンスの管理を説明する図である。
【符号の説明】
１－１，１－２　クライアント，　２　インターネット，　３　コンテンツサーバ，　４　ライセンスサーバ，　５　課金サーバ，　２０　タイマ，　２１CPU，　２４　暗号化復号部，　２５　コーデック部，　２６　入力部，　２７出力部，　２８　記憶部，　２９　通信部

【図１】

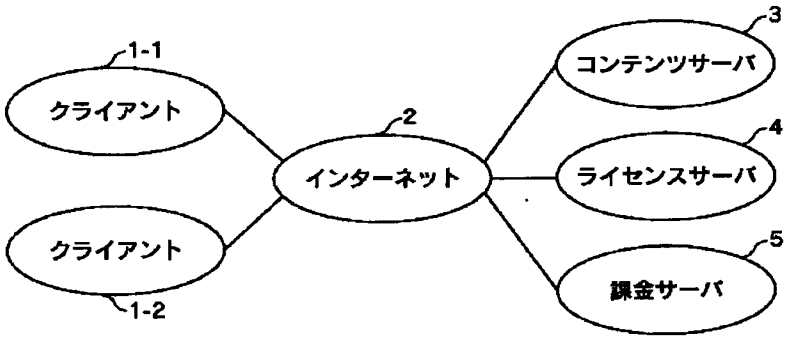


図1

【図３】

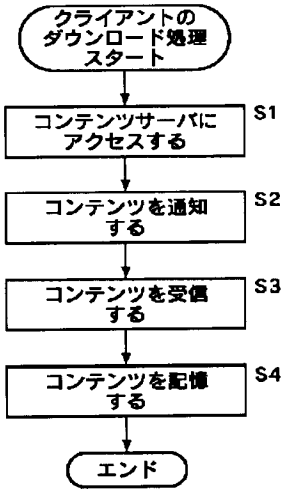


図3

【図８】

ライセンスID
作成日時
有効期限
使用条件
リーフID
電子署名

ライセンス

図8

【図２０】

EKB

Enc(DNK, KR)

図20

【図２】

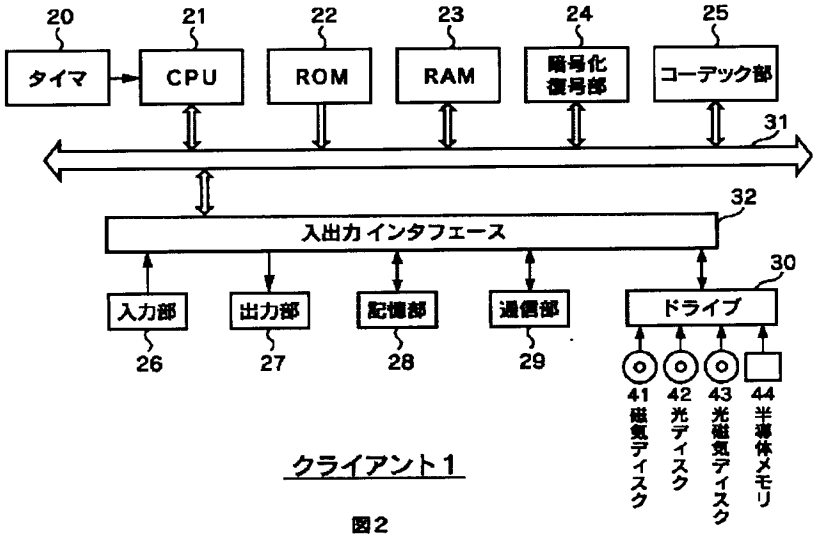


図２

【図４】

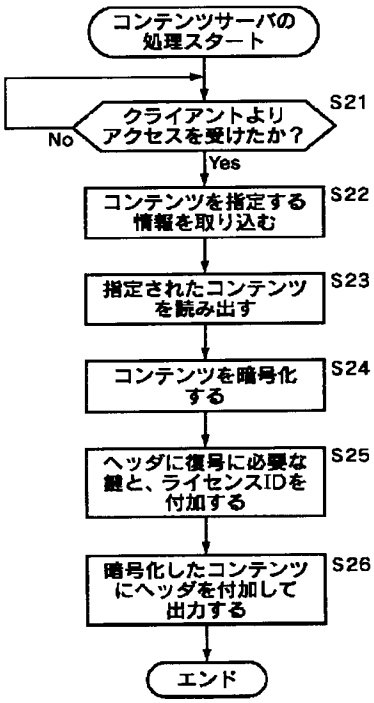


図４

【図５】

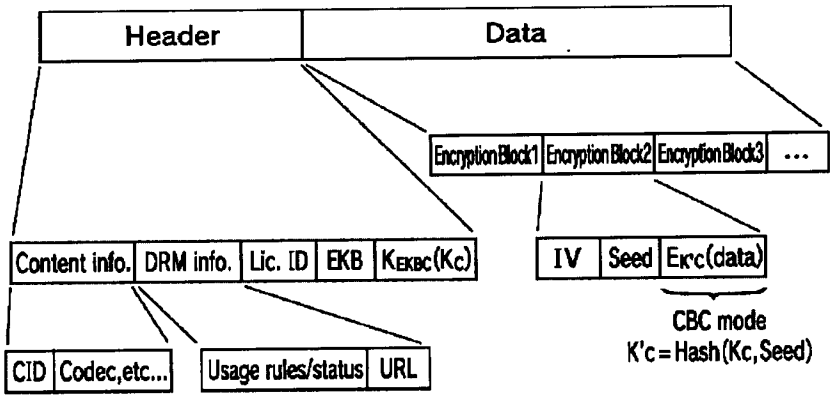


図５

【図１１】

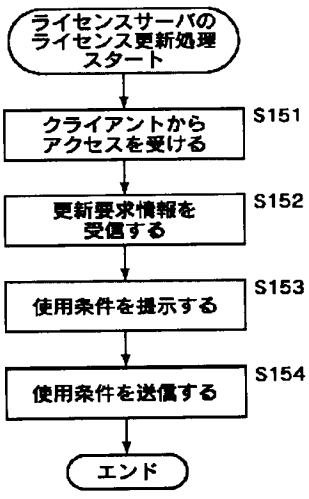


図１１

【図６】

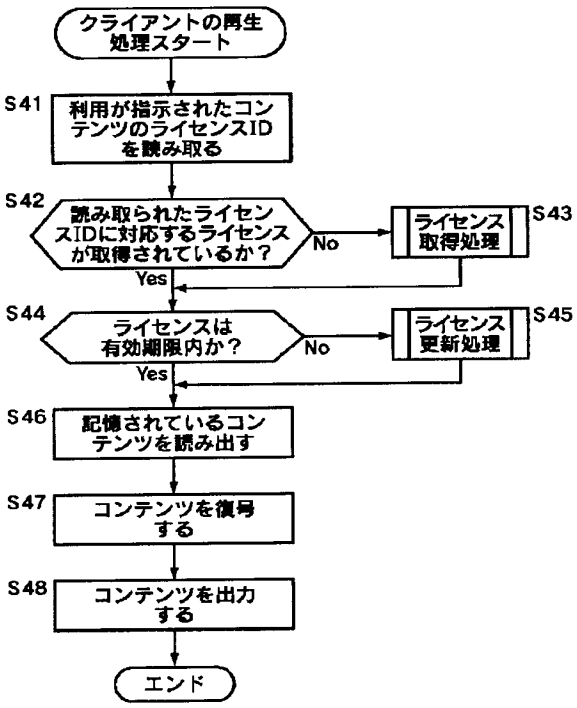


図６

【図７】

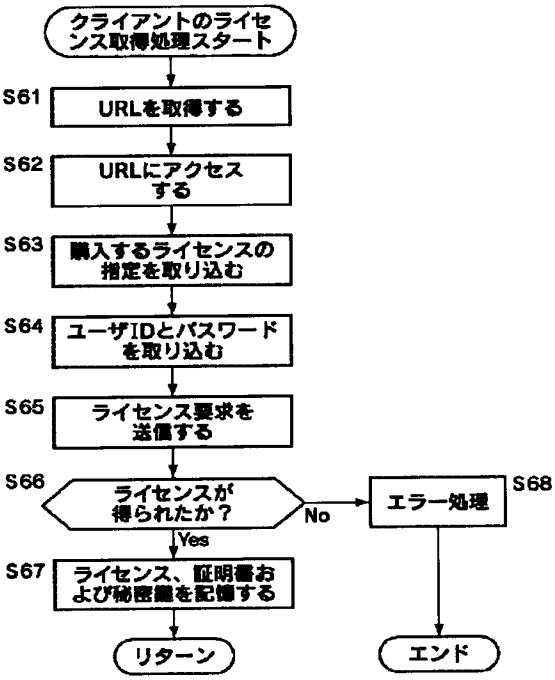


図７

【図１２】

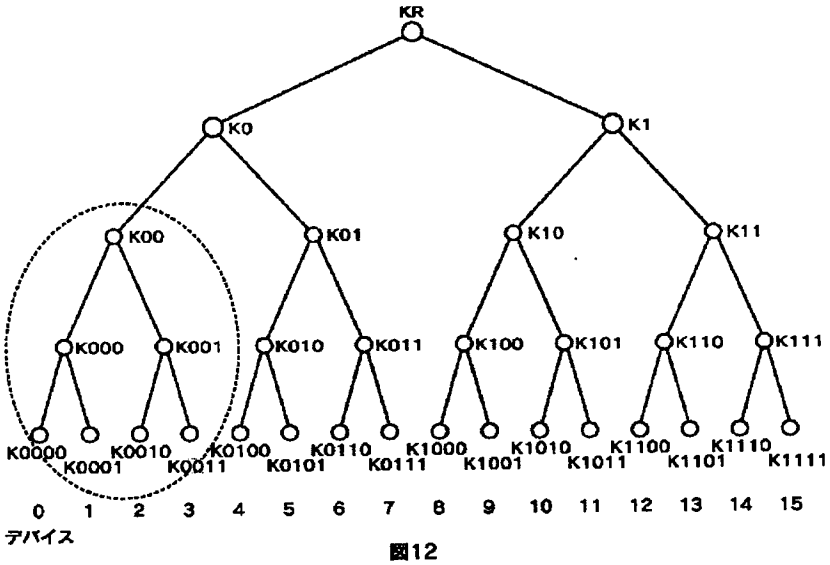


図12

【図２３】

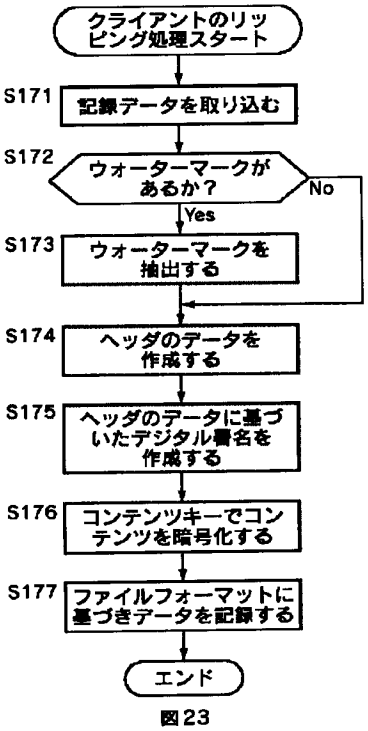


図23

【図９】

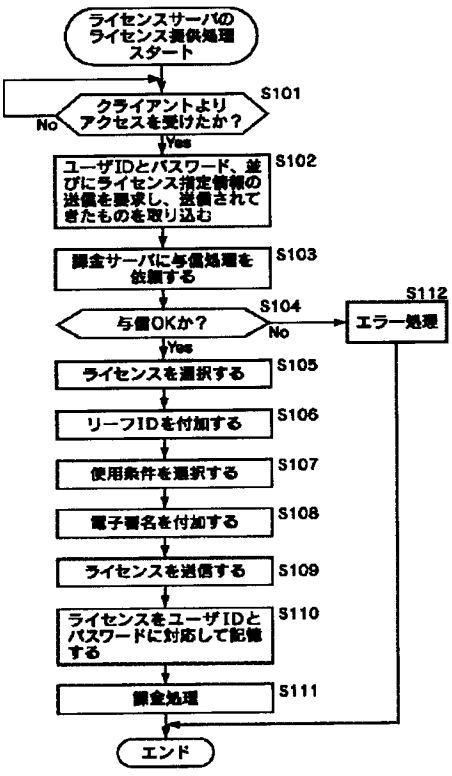


図9

【図１０】

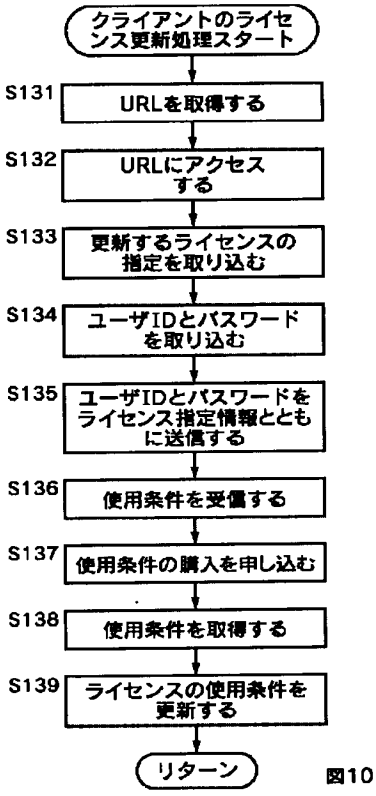


図10

【図２５】

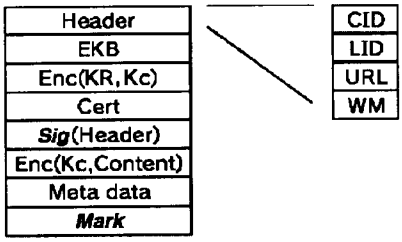


図25

【図３１】

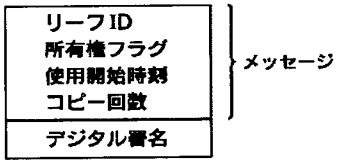


図31

【図１３】

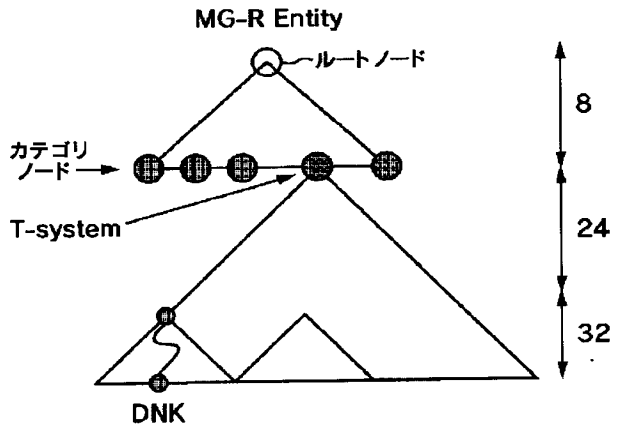


図13

【図３４】

$$Mark = \{LeafID, Own, Sig_S(LeafID, Own)\}$$

図34

【図１５】

A 有効化キーブロック(EKB:Enabling Key Block)
デバイス0,1,2にバージョン:tのノードキーを送付

バージョン(Version) : t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

B 有効化キーブロック(EKB:Enabling Key Block)
デバイス0,1,2にバージョン:tのノードキーを送付

バージョン(Version) : t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

図15

【図１４】

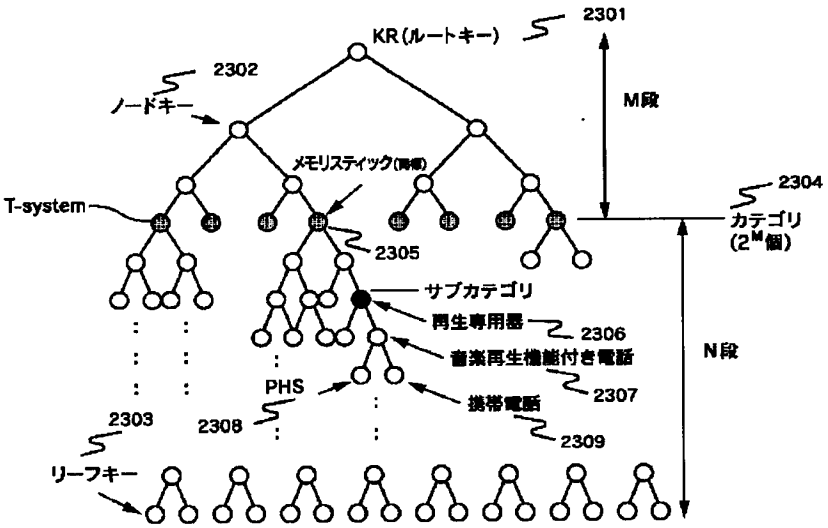


図14

【図１６】

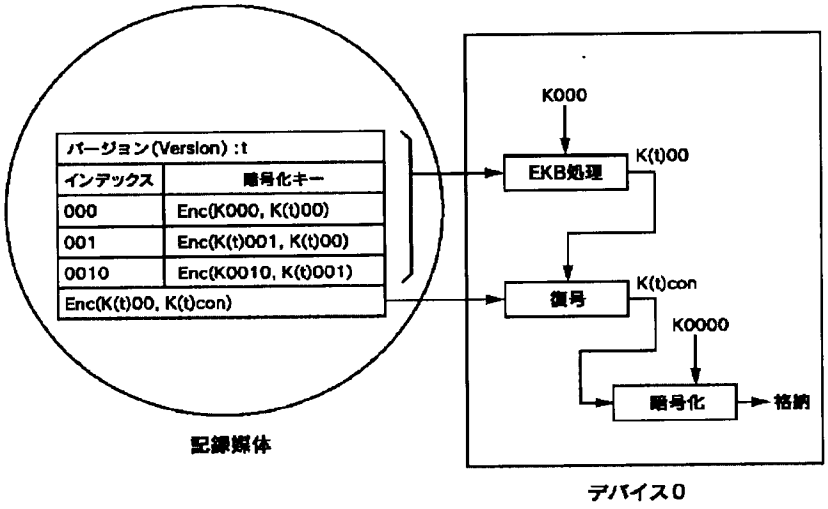


図16

【図３９】

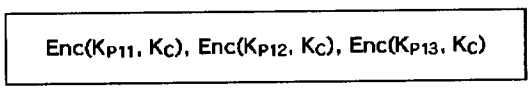


図39

【図３３】

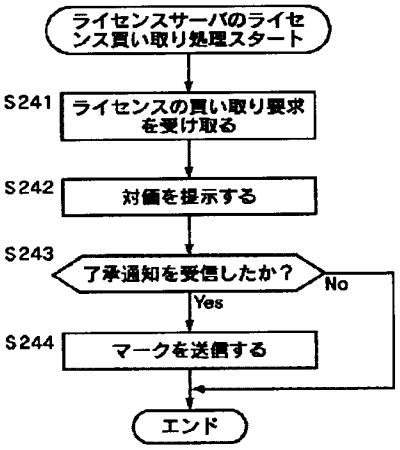


図33

【図３５】

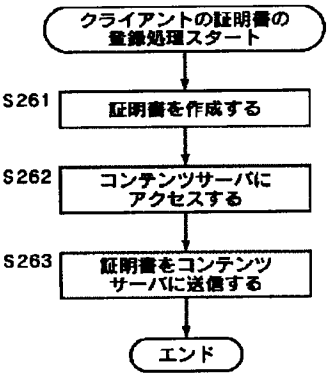


図35

【図３６】

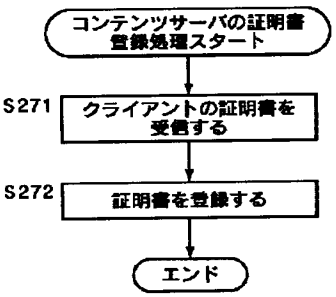
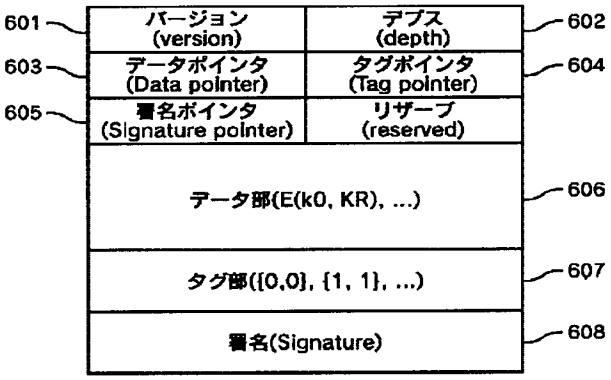


図36

【図１７】



EKB

図17

【図１９】

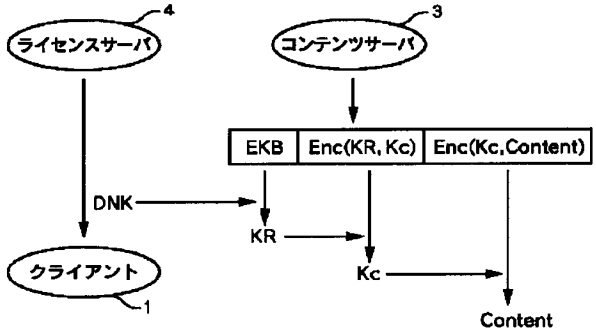


図19

【図１８】

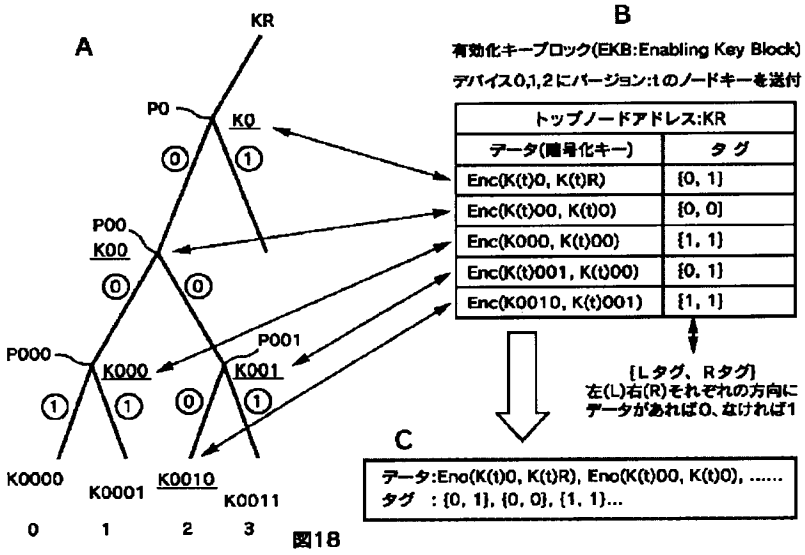


図18

【図３８】

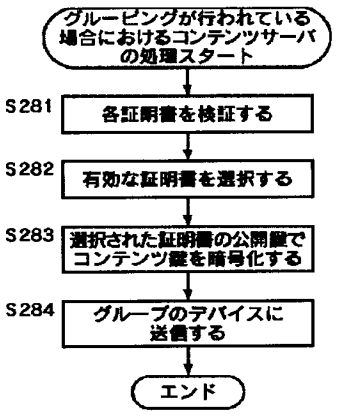


図38

【図４２】

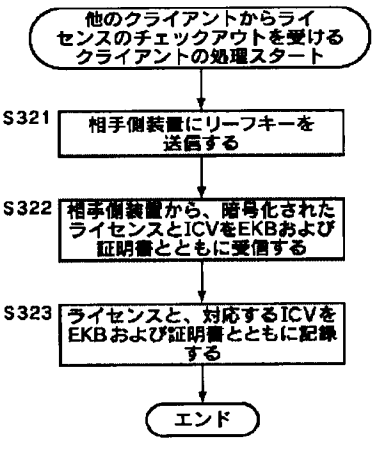


図42

【図３７】

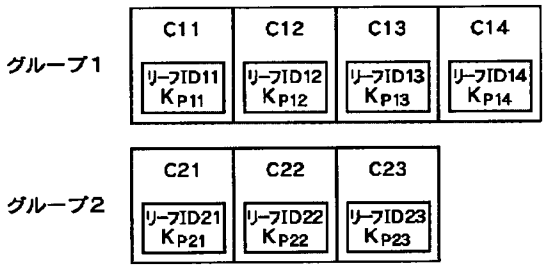


図37

【図４０】

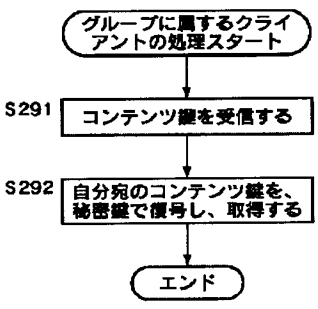


図40

【図２１】

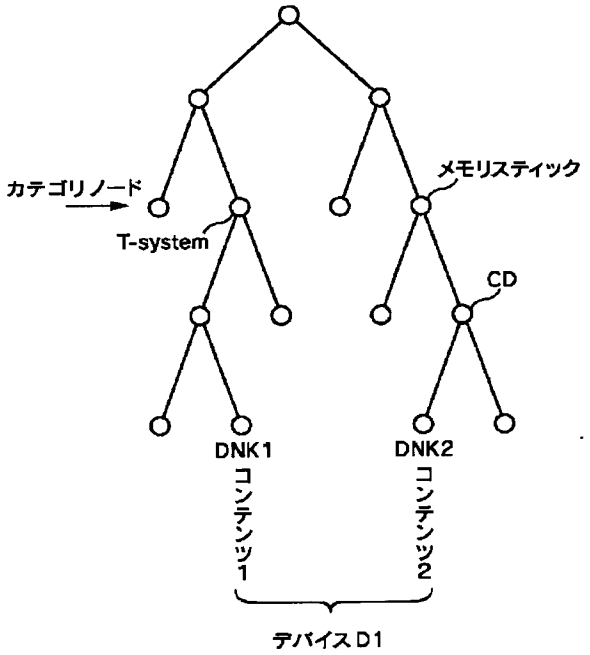


図21

【図２４】

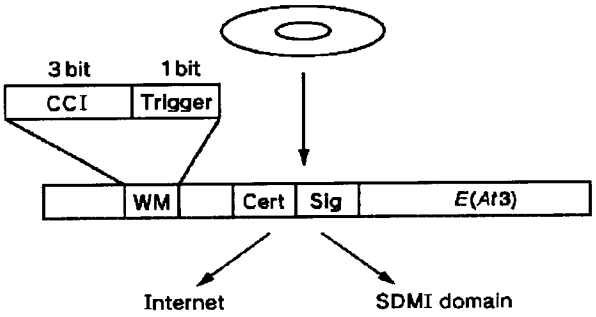


図24

【図２２】

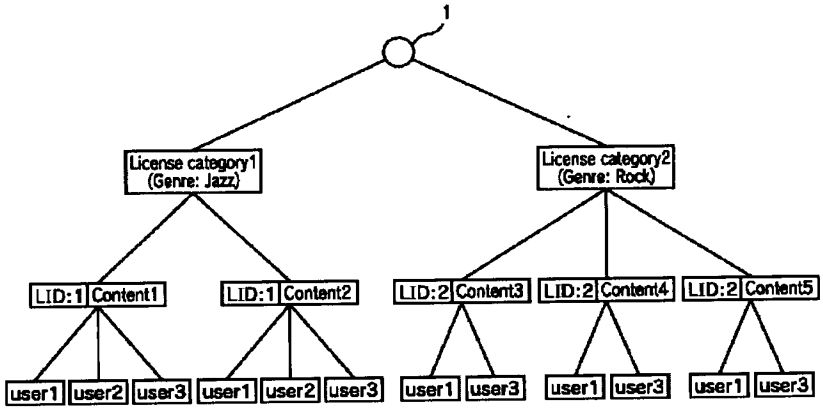
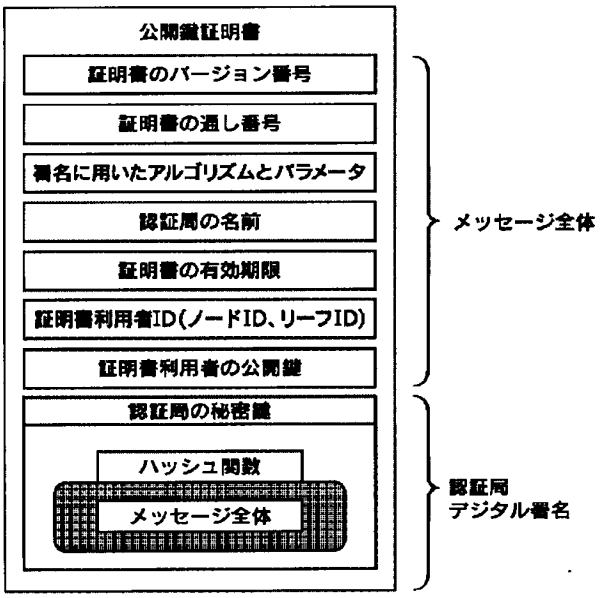


図22

【図２６】



公開鍵証明書

図26

【図２７】

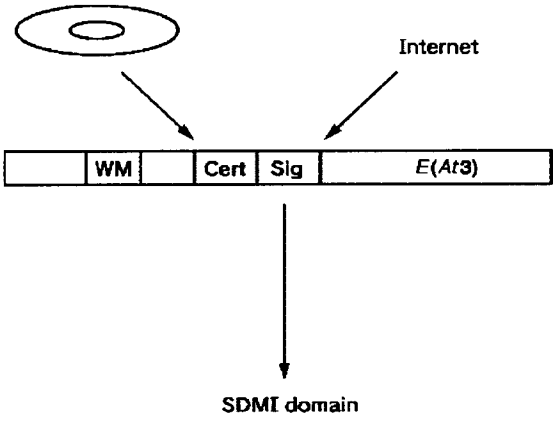


図27

【図２８】

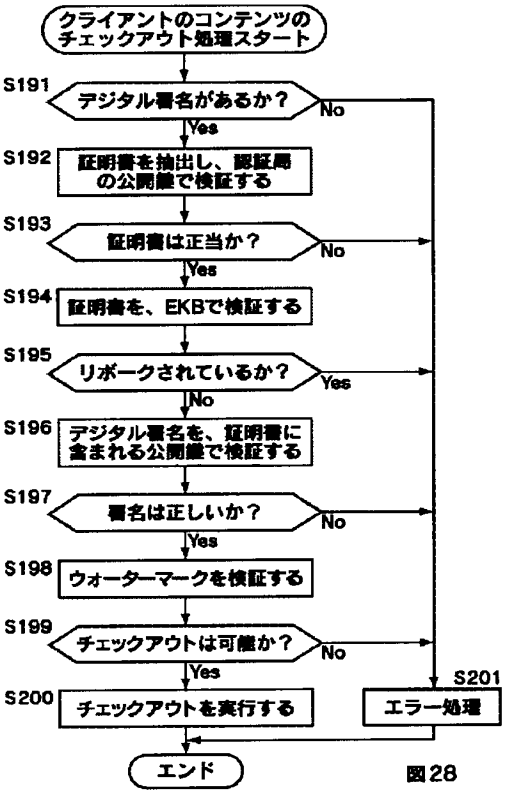


図28

【図２９】

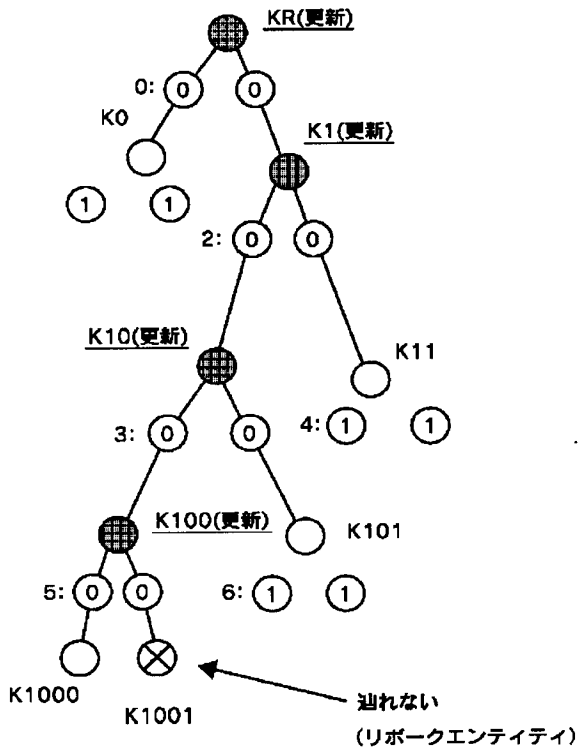


図29

【図３０】

有効化キーブロック(EKB:Enabling Key Block)の
データ部およびタグ

データ (暗号化キー)	Enc(K0, K(t)R), Enc(K(t)1, K(t)R) Enc(K(t)10, K(t)1), Enc(K11, K(t)1) Enc(K(t)100, K(t)10), Enc(K101, K(t)10) Enc(K1000, K(t)100)
タグ	0: [0, 0], 1:[1, 1], 2:[0, 0], 3:(0, 0) 4: [1, 1], 5:[0, 1], 6:[1, 1]

{Lタグ、Rタグ}
左(L)右(R)それぞれの方向に
データがあれば0、なければ1

図30

【図３２】

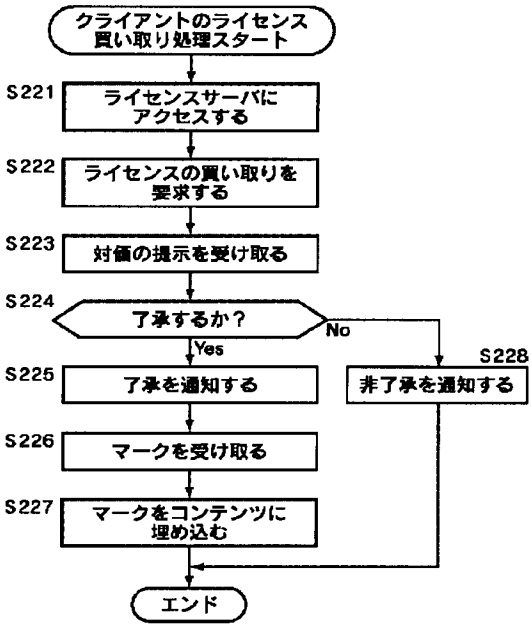


図32

【図４１】

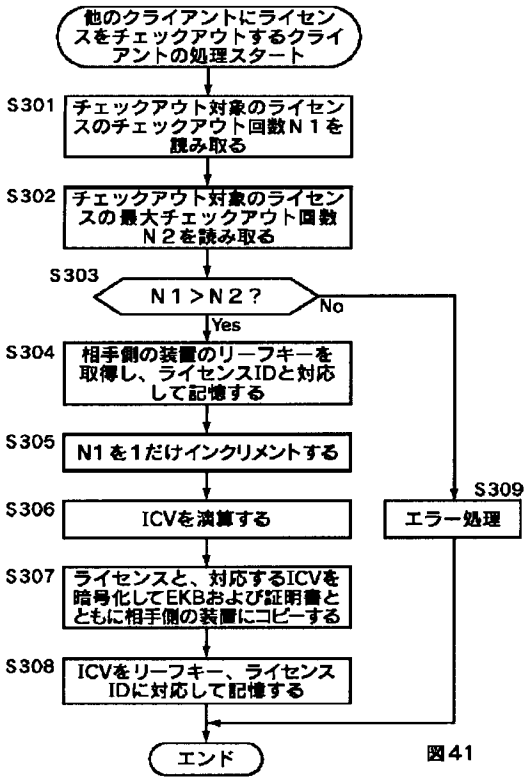


図41

【図４３】

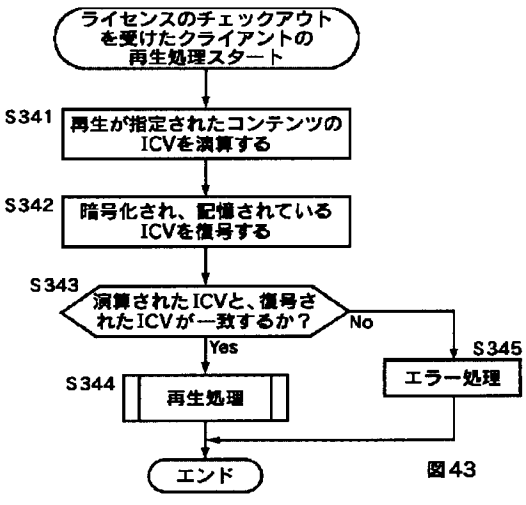


図43

【図４４】

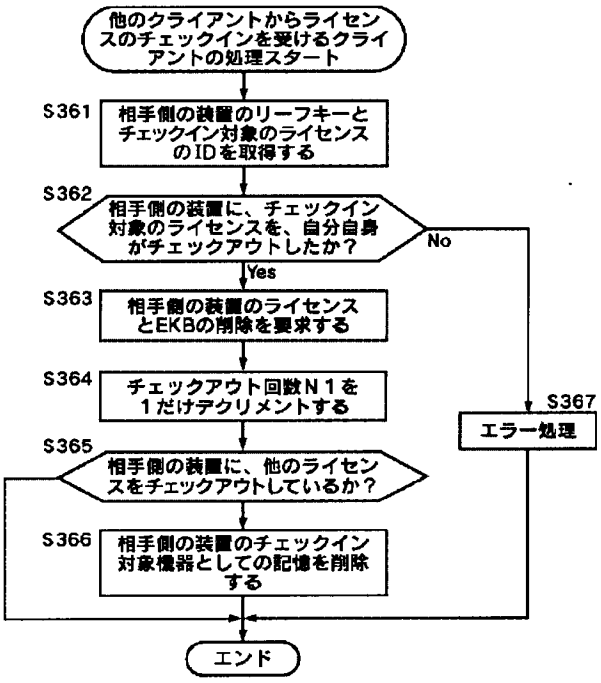


図44

【図４５】

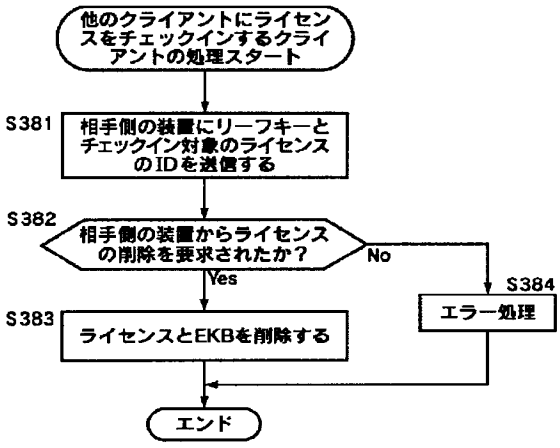


図45

【図４６】

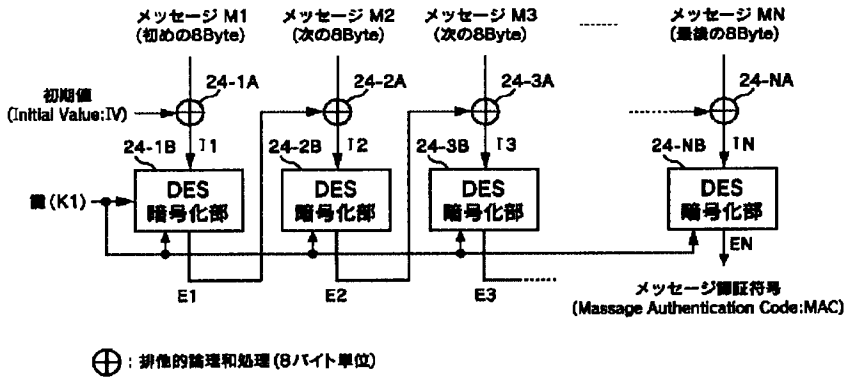


図46

【図４７】

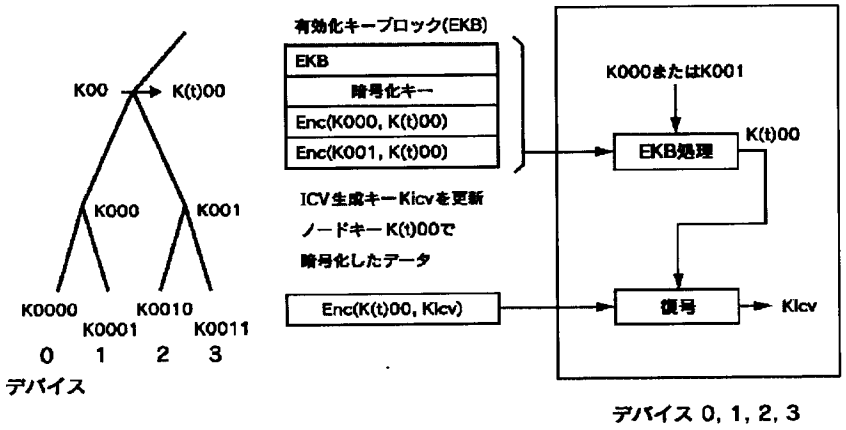


図47

【図４８】

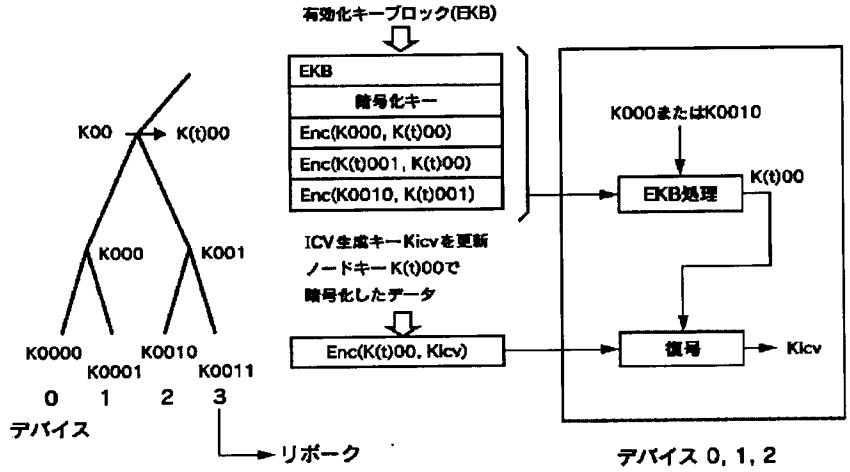


図48

【図５０】

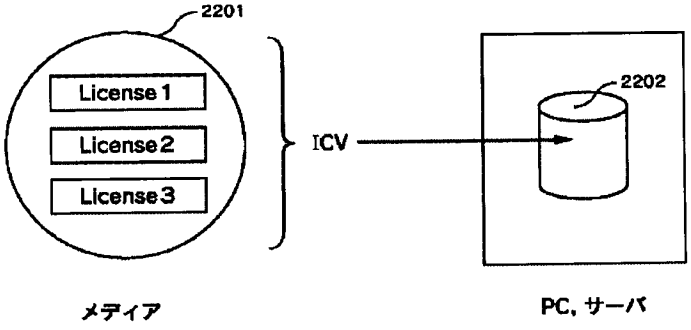


図50

【図４９】

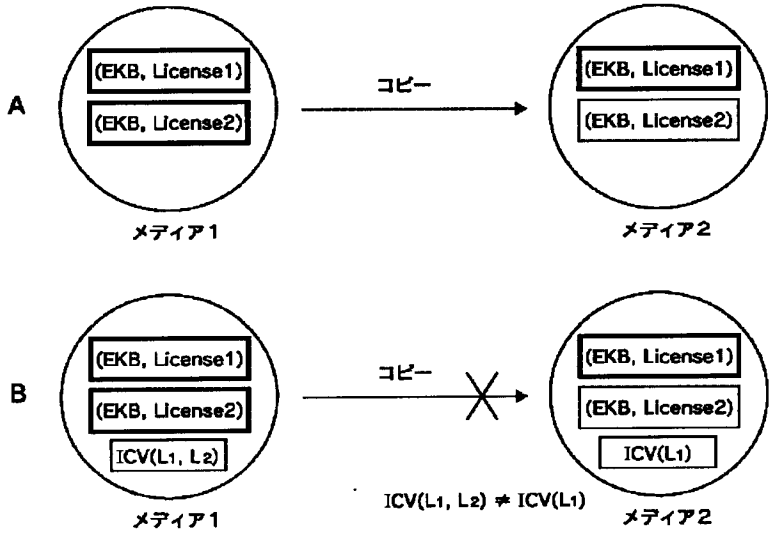


図49

フロントページの続き

(72)発明者　黒田　壽祐
東京都品川区北品川６丁目７番35号　ソニー株式会社内

(72)発明者　石黒　隆二
東京都品川区北品川６丁目７番35号　ソニー株式会社内

Ｆターム(参考)　5J104 AA12 MA05 PA07 PA10